

# Enhancing Security and Privacy in Video Surveillance through Role-Oriented Access Control Mechanism

**Mahmood Rajpoot, Qasim**

*Publication date:*
2016

*Document Version*
Publisher's PDF, also known as Version of record

# Enhancing Security and Privacy in Large-Scale Video Surveillance through Role-Oriented Access Control Mechanism

Qasim Mahmood Rajpoot


Kongens Lyngby 2016
PhD-2016-399

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk PhD-2016-399

# Summary

Use of video surveillance has significantly increased in the last few decades. Modern video surveillance systems are equipped with techniques that automatically extract information about the objects and events from the video streams and allow traversal of data in an effective and efficient manner. Pervasive usage of such systems gives substantial powers to those monitoring the videos and poses a threat to the privacy of anyone observed by the system. Aside from protecting privacy from the outside attackers, it is equally important to protect the privacy of individuals from the inside personnel involved in monitoring surveillance data to minimize the chances of misuse of the system, e.g. voyeurism. In this context, several techniques to protect the privacy of individuals, called privacy enhancing techniques (PET) have therefore been proposed in the literature which detect and mask the privacy sensitive regions, e.g. faces, from the videos. However, very few research efforts have focused on addressing the security aspects of video surveillance data and on authorizing access to this data. Interestingly, while PETs help protect the privacy of individuals, they may also hinder the usefulness of video surveillance systems resulting in compromising the very purpose of such systems, i.e. public safety. Thus the challenge is to provide sufficient need-specific data to those monitoring the surveillance systems yet preserving the privacy of people as much as possible. This can be achieved through a dynamic access control mechanism that may provide proportionate access to data while allowing reversing the PETs whenever required. In this context, a summary of thesis contributions is given below.

In this thesis, we present an abstract model of video surveillance systems that helps identify the major security and privacy requirements in a video surveillance system. We study existing solutions against these requirements and point out practical challenges in ensuring the security of video surveillance data in all

stages (in transit and at storage). Our study shows a gap, between the security requirements that we identified and the proposed security solutions, where future research efforts may focus in this domain. From the challenges that we outline regarding security in video surveillance, we focus on development of a dynamic access control mechanism.

We develop a general-purpose access control model that is suitable for video surveillance systems as well as other domains sharing similar requirements. As the currently dominant access control models – the role-based access control (RBAC) and the attribute-based access control (ABAC) – suffer from limitations while offering features complementary to each other, their integration has become an important area of research. Our access control model combines the two models in a novel way in order to unify their benefits while avoiding their limitations. Our approach provides a mechanism that not only takes information about the current circumstances into account during access control decision making, but is also suitable for applications where access to resources is controlled by exploiting the contents of resources in the access control policy. We evaluate our model against RBAC and ABAC and demonstrate that our model brings together the benefits offered by RBAC and ABAC while addressing the role- and permission-explosion issues faced in RBAC.

Based on our access control model, we then present an access control mechanism for video surveillance systems. Contrary to the existing approaches, the proposed access control mechanism is role-oriented and retains advantages associated with role-based access control, yet it allows specification of policies using the metadata associated with the objects as well as the attributes of users and environment. In addition to role hierarchies, the content-based permissions in our model allow derivation of several permissions from the explicitly stated ones due to the hierarchical relations between the attributes of different entities. We implement a prototype of the proposed mechanism and demonstrate that the access control policies using our approach may be specified via eXtensible Access Control Markup Language (XACML).

# Resumé

Anvendelse af videoovervågning er steget betydeligt i de seneste årtier. Moderne videoovervågningssystemer er udstyret med teknikker, der automatisk udtrække oplysninger om objekter og hændelser fra video strømme og tillade gennemgang af data på en effektiv måde. Allestedsnærværende brug af sådanne systemer giver omfattende beføjelser til dem der overvåge videoerne og det udgør en trussel mod privatlivets fred for enhver der observeres af systemet. Bortset fra at beskytte personfølsomme oplysninger imod udefrakommende angribere, er det lige så vigtigt at beskytte personfølsomme oplysninger fra det indvendige personale, der deltager i overvågningen af video data for at minimere risikoen for misbrug af systemet, f.eks voyeurisme. I denne sammenhæng er flere teknikker til beskyttelse af personfølsomme oplysninger, kaldet privacy enhancing techniques (PET), derfor blevet foreslået i litteraturen. Disse teknikker registrerer og maskere personfølsomme oplysninger i video strømme, f.eks ansigter. Imidlertid har meget få forskningsindsatser fokuseret på at løse de sikkerhedsmæssige aspekter af data videoovervågning og om at give adgang til disse data. Imens PETer hjælper med at beskytte personfølsomme oplysninger, kan de også hindre nytten af videoovervågningssystemer, dette resulterer i kompromittering af formålet med sådanne systemer, dvs. den offentlige sikkerhed. Således er udfordringen at give tilstrækkelig behov-specifikke data til dem der overvåger disse systemer, mens de personfølsomme oplysninger beskyttes så meget som muligt. Dette kan opnås gennem en dynamisk adgangskontrol, der kan give et forholdsmæssig adgang til data, samtidig med at vende effekten af PETer når det er påkrævet. Nedenfor gives et sammendrag af afhandlingens bidrag.

I denne afhandling præsenterer vi en abstrakt model af videoovervågningssystemer, der hjælper med at identificere de væsentligste sikkerheds krav og privacy krav i et videoovervågningssystem. Vi studerer eksisterende løsninger mod disse

krav og påpege praktiske udfordringer i at garantere sikkerheden for video overvågningsdata i alle faser (i transit og ved opbevaring). Vores studie viser en kløft, mellem de sikkerheds krav som vi identificerede og de foreslåede sikkerhedsløsninger, hvor fremtidige forskningsindsatser kan fokusere på dette område. Fra de udfordringer, som vi har fremført med hensyn til sikkerheden i videoovervågning, fokuserer vi på udvikling af en dynamisk adgangskontrolmekanisme.

Vi udvikler en generel adgangskontrol model, der er egnet til videoovervågningssystemer samt andre domæner, med lignende krav. Da de aktuelt dominerende adgangskontrol modeller – role-based access control (RBAC) og attribute-based access control (ABAC) - lider af begrænsninger mens de tilbyder funktioner der supplerer hinanden, er deres integration blevet et vigtigt forskningsområde. Vores adgangskontrol model kombinerer de to modeller i en ny måde for at forene deres fordele og samtidig undgå deres begrænsninger. Vores tilgang indeholder en mekanisme, der ikke kun tager oplysninger om de nuværende omstændigheder i betragtning under adgangskontrol beslutningstagningen, men er også velegnet til applikationer, hvor adgang til ressourcer styres ved at udnytte indholdet af ressourcerne i adgangskontrol politiken. Vi evaluerer vores model mod RBAC og ABAC og vise, at vores model samler fordelene ved RBAC og ABAC samtidig med at de adressere role- og permission-explosion udfordringerne i RBAC.

Baseret på vores adgangskontrol model, præsentere vi en adgangskontrol mekanisme for videoovervågningssystemer. I modsætning til de eksisterende tilgange, er den foreslåede adgangskontrol mekanisme role-oriented og bevarer fordele forbundet med role-based access control, men det giver mulighed for specifikation af politikker ved hjælp af metadata i forbindelse med de ressourcer samt attributterne for brugere og miljø. Udover role hierarkier, tillader de indholdsbaserede permissions i vores model afledning af flere permissions fra de udtrykkeligt betegnede, på grund af de hierarkiske relationer mellem attributterne for forskellige enheder. Vi implementerer en prototype af den foreslåede mekanisme og vise, at de adgangskontrolpolitikker ved hjælp af vores tilgang kan specificeres via eXtensible Access Control Markup Language (XACML).

# Preface

This thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark (DTU Compute) in partial fulfillment of the requirements for acquiring the degree of Doctor of Philosophy.

The thesis attempts to address the security and privacy issues in large-scale video surveillance systems. The thesis is self-contained and revolves around the work done in a number of publications written during the period 2013-2016.

The PhD project has been supervised by Associate Professor Christian Damsgaard Jensen and was carried out in close collaboration with Milestone Systems, Denmark and Dr. Ram Krishnan of The University of Texas at San Antonio, USA.

The work has been supported by a grant from the Danish National Advanced Technology Foundation (DNATF – "Højteknologifonden", which is now integrated with the Innovation Foundation).

Lyngby, May-2016

Qasim Mahmood Rajpoot

# Dedication

*To my parents:*
*for their unconditional love, support and numerous sacrifices which enabled me to accomplish what I have achieved today*

# Acknowledgments

I would like to express my sincere thanks to my advisor Dr. Christian Damsgaard Jensen for providing me this opportunity and for his invaluable guidance and extraordinary patience. He has always been very supportive and encouraging and provided me the freedom to pursue new ideas. I am also very grateful to Dr. Ram Krishnan, from University of Texas at San Antonio (UTSA), USA. I have greatly benefited from his guidance during my research stay at UTSA. I would also like to express my thanks to Søren Løfvall, Max Ottosen and Morten Boysen from Milestone Systems, Denmark for their guidance and support. The fruitful discussions I had with them helped me stay focused and to develop a practical solution for the problem I worked on.

Thanks also goes to Karin Tunder for her endless support with administrative matters and Rasmus Sørensen for his help in writing the Danish summary of this thesis. I would also like to thank my friends and colleagues at DTU for creating an inspiring environment and for making my stay at DTU a truly enjoyable experience.

I am particularly thankful to my family: my brothers and sisters for their love, support and cheering me up in difficult times. Special thanks to my elder brother Nasir Rajpoot for his invaluable guidance and exceptional support without which I would not have achieved this milestone. My daughter, Maryam, who is a constant source of joy and whose giggles and laughs kept me going during the most demanding times of my life. Last but not the least, I would like to thank my loving wife, Ume Habiba. This thesis would not have been possible without her encouragement and extraordinary support.

# List of Papers

The following publications were the result of the research work described in this thesis:

- [113] Rajpoot, Q. M., and Jensen, C. D. Security and privacy in video surveillance: Requirements and challenges. In *29$^{th}$ IFIP International Information Security and Privacy Conference (IFIP-SEC)*, 2014, Springer, pp. 169–184. Published.
- [114]: Rajpoot, Q. M., and Jensen, C. D. Video surveillance: Privacy issues and legal compliance. *Promoting Social Change and Democracy Through Information Technology*, V. Kumar and J. Svensson, Eds. IGI Global, 2015, pp. 69–92. Published.
- [117]: Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Integrating attributes into role-based access control. In *29$^{th}$ Data and Applications Security and Privacy Conference (DBSec)*, 2015, Springer, pp. 242–249. Published.
- [116]: Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Attributes enhanced role-based access control model. In *12$^{th}$ International Conference on Trust, Privacy and Security in Digital Business (TrustBus)*, 2015, Springer, pp. 3–17. Published.
- [115]: Rajpoot, Q. M., and Jensen, C. D.: Role-oriented Access Control Model for Video Surveillance Systems. *Elsevier Computers & Security* (To be submitted).

# Contents

# List of Figures

# List of Tables

Chapter 1

# Introduction and Motivation

The work presented in this thesis is part of a large project, Managed Video as a Service (MVaaS), that involves four PhD students working on following four aspects of video surveillance systems: data extraction, data indexing, architecture, and security. The work has been carried out in close collaboration with Milestone Systems, Denmark which is one of the leading IP-based video surveillance solution providers. The goal of the MVaaS project is to support the vision of Milestone Systems with respect to future video surveillance systems, where users will require ubiquitous access to live and stored video data. In this thesis, we focus in particular on the security and privacy challenges arising from this vision.

## 1.1 Video Surveillance

The use of video surveillance has increased manifold in the last two decades, particularly in the developed countries [46]. Continuous security threats to public safety and an increased sense of insecurity caused by incidents of terrorism, such as the 9/11 attacks, the London and Madrid bombings and the recent incidents in France and elsewhere, demand to build infrastructure to protect against harm

to people and property. The rapidly decreasing costs of hardware including cameras, storage and communication infrastructure are also playing an important role in the ubiquitous spread of video surveillance systems [130]. As the cost of hardware and communication infrastructure has come down while the cost of security personnel's training, management and salaries has generally increased, considerable savings can be achieved through incorporation of video surveillance systems. Therefore, a video surveillance system is considered an invaluable and indispensable tool in combating crime. Law enforcement agencies worldwide rely on these systems for helping to prevent, detect and investigate attacks against public safety. As a consequence, in many developed countries, surveillance cameras are now frequently found in office buildings, shopping malls, housing estates, streets, squares, parks, buses, trains, stations, airports and various other public places.

Traditional video surveillance systems, also known as closed-circuit television (CCTV), are simple recording systems which need to be monitored by humans without automated technological assistance. This makes them very expensive in terms of operation due to the tedious and time consuming process of manually watching videos. They are mainly used as deterrents and the recordings help the investigation once an incident has occurred. Compared to these traditional solutions, modern digital solutions are less expensive while offering much better quality. Modern systems are equipped with advanced techniques such as object-detection, object-identification, object-tracking and event-detection, by exploiting algorithms from the fields of computer vision, image processing and pattern recognition [58]. These techniques potentially allow those monitoring the videos to recognize a target object e.g. a vehicle, or even automatically tracking an individual spanning over multiple areas in a surveillance network [87], with trivial effort.

## 1.2 Privacy Protection

The deployment of surveillance cameras all around major public areas is a source of concern due to its impact on the privacy of involved individuals. Privacy advocates and civil libertarians consider video surveillance a serious threat to the privacy of non-criminals who may be captured by cameras in public places several times a day [76, 134]. The use of pervasive video surveillance may lead to a "big brother" society perception in which all the activities of an individual can be profiled, either allowed legally by law enforcement authorities or performed out of curiosity by a personnel. Doing so requires a significant amount of time and effort in traditional surveillance systems. Yet there have been reported incidents, in traditional video surveillance, where the guards observing videos

were involved in unauthorized collection of data on the activities of individuals [19, 34, 141]. For instance, in a report by BBC News [19], a group of council employees in the UK spied on a woman's apartment using surveillance cameras installed in that area. The possibilities for such misuse are further increased with the advent of modern video surveillance systems that facilitate rapid data retrieval enabled by searching and advanced imaging technology. Thus, the privacy concerns are obviously much more serious in modern video surveillance systems compared to traditional ones.

Despite these privacy concerns, the importance of video surveillance systems in combating the security threats is considered very important [49]. As criminals and terrorists increasingly make use of new technology to mount attacks on public safety, the public and law enforcement agencies must continuously increase their technological capabilities to protect innocent citizens. However, the need for increased security does not imply any less importance for privacy. In order to increase the public acceptance of video surveillance systems, it is important that their deployment strikes a balance between security and the need to protect privacy. Therefore, surveillance systems must be designed and used in ways that protect individuals against crime, without compromising their rights to privacy.

The need to balance the usage of video surveillance against its negative impacts has been the focus of several research efforts during the last few years. It is important to note that in video surveillance, often it is the behavioral information of people – activity of people – which is important to monitor rather than the identity of the people [34]. In this context, various privacy enhancing techniques (PETs) have been proposed in order to protect the privacy of observed people by detecting and masking the privacy sensitive regions of people, e.g. faces, in images and captured video. Some of the key categories of PETs include obfuscation [127, 122, 160], scrambling [27, 32, 46], and abstraction [61, 73, 130]. Obfuscation reduces the level of details of privacy sensitive areas with the help of blurring or pixilation, scrambling encrypts the sensitive regions with a key allowing the area to be decrypted only by authorized personnel, and abstraction replaces the semantic objects, e.g. humans, in the video with dummy objects such as silhouettes or skeletons.

## 1.3 Security in Video Surveillance

Due to the use of digital video surveillance where data is transferred over the Internet, security of data is an important requirement because of the stringent need to ensure confidentiality, integrity, and availability of information. A comprehensive solution covering all aspects of security is quite complicated and

requires the integration of different tools and techniques. In such a context, access control plays a fundamental role by establishing which users are authorized to perform what actions on which objects. Many solutions addressing the security requirements including integrity [14, 128, 137], confidentiality [80, 82, 133] and access control [24, 23, 107] have been proposed in multimedia systems, e.g. video on demand and business video conferencing. However, key factors involved in video surveillance systems are quite different from those in multimedia systems. Hence these solutions cannot be directly applied in video surveillance systems, though a few commonalities exist. In video surveillance systems, unlike multimedia applications, there are several video producers (cameras) with limited processing capabilities. Due to the communication over public networks, the security aspect is to be addressed when data is transferred from camera-to-server, server-to-server and server-to-handheld devices or monitoring room. Moreover, in video surveillance, there is a unique requirement to protect privacy of individuals by obfuscating the privacy sensitive regions. However, our study of literature reveals that little research attention has been paid to address the security of video streams and the associated data while they are transmitted or stored. Similarly, little research is found in the literature that targets the challenge of access control in video surveillance systems. This thesis focuses on access control and presents our work to enhance security and privacy in video surveillance systems through a dynamic access control mechanism.

### 1.3.1 Access Authorization

Modern video surveillance systems normally employ a network of several cameras deployed throughout the surveilled region to capture video data at their respective locations, and transfer this data to the storage servers and the users via public networks. Users wishing to access live or recorded data may either access it in a monitoring room, similar to traditional CCTV systems, or using a computing equipment, such as their hand-held devices. Such users are commonly referred to as *observers* in video surveillance. As the potential capabilities offered by modern video surveillance systems – such as searching for an individual or a semantic object contained in video, and monitoring the activities of an individual spanning over multiple locations – make it easy to invade an individual's privacy, it becomes critically important to control the access to data in such systems [87]. Clearly, video surveillance is expected to become more pervasive which leaves us with only two choices: either trust all the observers or devise a mechanism for watching the watchers in order to minimize the chances of using such systems abusively [122].

On the one hand, PETs help protect the privacy of people but on the other hand, they might impede the efficacy of a video surveillance system. For instance, in

case of an anomalous incident, the observers may need to access full information in a video, e.g. revealing the identity of a person, in order to closely investigate the incident. This interesting observation implies that the existing PETs cannot be effectively applied until there is a mechanism to reverse these PETs when required. Therefore, the use of PETs without considering current circumstances may render the system useless as an observer may be disallowed full access to the data when needed. Thus, the challenge is to utilize the video surveillance system by exposing sufficient need-specific data while simultaneously preserving the privacy [87].

Video data contains multiple levels of information which may include original raw video, reconstructed video with regions revealing identity removed deliberately, e.g. blurred faces, video with lower resolution, etc. Different users can be given access to varying granularity levels of information in the video depending on their authorizations and the current circumstances. In normal circumstances, an observer may be granted access to only the behavioral information while hiding the privacy-sensitive regions. However, in case of an emergency, full access to the videos may be authorized. One possible way to protect the privacy in video surveillance, while retaining the useful functionality of video surveillance, is to use a dynamic access control mechanism that utilizes the plethora of PETs in video surveillance systems. A dynamic access control mechanism may enable preserving the privacy of people yet allowing maximum need-specific access to the data by providing multiple privacy levels, with each level accessible to different access privileges in different circumstances.

As mentioned above, very few research efforts have focused on access control in video surveillance systems. One reason for lack of research focus in this area is because it is dependent on certain related research areas: i) efficient video analysis for semantic visual concept representation, called video analytics, and ii) effective indexing structure linking the semantic visual concepts with the videos to enable efficient retrieval [23]. Recently, several research efforts targeting extraction of semantic concepts – semantic objects e.g. humans, vehicles, etc., and semantic events e.g. vandalism, fire, luggage left-behind, etc. – contained in the video, have been reported; see for instance [16, 109, 146]. These semantic concepts are stored as part of the metadata associated with a video, in order to allow retrieval of videos based on metadata information. Various solutions to retrieve videos based on metadata information have been designed [59, 144, 148]. An ISO/IEC standard, MPEG-7 [86], has also been developed, focusing exclusively on associating metadata with the video and providing the basis to retrieve videos based on metadata. To achieve efficient retrieval of videos, solutions to store MPEG-7 descriptions in the form of multimedia databases have also been proposed [17, 44, 149]. Metadata information associated with the video may also be used in specification of the access control policy in video surveillance systems. For example, a user may be authorized to access all videos belonging

to a specific region which contain a truck and were recorded during 1700 – 1900 hrs. Such content-dependent authorization on video data allows granting or denying access to videos based on the information within the video data. With these advancement in the areas of video analytics and indexing, effective access control mechanisms for video surveillance systems may now be designed.

### 1.3.2 Access Control Paradigms

Role-based access control (RBAC) and attribute-based access control (ABAC) are the most popular access control paradigms today [38, 74]. Researchers have shown that traditional access control models (cf. § 2.4), including discretionary access control and mandatory access control (MAC), can also be configured through these paradigms [66, 106]. The RBAC paradigm encapsulates privileges into roles, and users are assigned roles to acquire privileges, which makes it simple to administer and facilitates reviewing permissions assigned to a user [38]. However, in RBAC, permissions are specified in terms of object identifiers, referring to individual objects. In many applications such as video surveillance, access to data is more naturally described in terms of its semantic contents [24] for example, a user may be granted access to a video in case there is unattended luggage left behind. Moreover, it has been recognized that RBAC is not adequate for situations where information about the current circumstances is a required parameter in granting access to a user [74]. A relatively new access control paradigm, ABAC [161] has been identified to overcome these limitations of RBAC [38]. However, ABAC is typically much more complex than RBAC in terms of policy review, hence analyzing the policy and reviewing or changing user permissions are quite cumbersome tasks. Both RBAC and ABAC have their particular advantages and disadvantages yet they both have features complimentary to each other, and thus integrating RBAC and ABAC has become an important research topic [38, 63, 67]. Also, the National Institute of Standards and Technology (NIST) has announced an initiative to integrate RBAC and its various extensions with ABAC in order to combine the advantages offered by both RBAC and ABAC [74].

We realize that video surveillance systems introduce several challenging requirements with respect to the formulation, specification and enforcement of appropriate access control policies. Among others, these requirements demand that the access control policy must be specified based on the semantic contents of data to be protected and the occurrence of anomalous incidents must be considered when granting an access request. In addition, the access control model needs to facilitate the visualization of any modifications done in the policy. Some of these requirements, such as content-based authorization, require the use of ABAC approach while others such as modification visualization require

using RBAC approach. Motivated by the access control requirements in video surveillance and the NIST initiative, we propose an access control model that combines the features offered by both RBAC and ABAC. Our work focuses on development of a novel content-based access control mechanism yet providing administrative benefits typically offered by RBAC.

## 1.4 Thesis Contributions

The main contributions of the work presented in this thesis and its appendices are summarized below:

1. We identify threats to privacy posed by video surveillance systems and investigate the legal infrastructure for ensuring privacy. We classify the video surveillance systems and the legislation in different countries that may apply on video surveillance systems. Further, we suggest guidelines in order to help those who want to deploy video surveillance while least compromising the privacy of people and complying with legal infrastructure. Parts of this work have been published as the following peer-reviewed book chapter:

- [114]: Rajpoot, Q. M., and Jensen, C. D. Video surveillance: Privacy issues and legal compliance. *Promoting Social Change and Democracy Through Information Technology*, V. Kumar and J. Svensson, Eds. IGI Global, 2015, pp. 69–92. Published.

2. We propose an abstract model of video surveillance to help identify a list of security and privacy requirements in a video surveillance system. We examine the existing solutions proposed to fulfill the major security and privacy requirements identified through our model and outline the associated challenges. Our study identifies a potential gap where research efforts need to be put in by pointing out challenges that need to be addressed while designing security solutions in this regard. This work resulted in the following publication accepted at a peer-reviewed conference:

- [113] Rajpoot, Q. M., and Jensen, C. D. Security and privacy in video surveillance: Requirements and challenges. In *29$^{th}$ IFIP International Information Security and Privacy Conference (IFIP-SEC)*, 2014, Springer, pp. 169–184. Published.

3. In our quest to develop an access control model for video surveillance systems, we build a general-purpose Attributes Enhanced Role-Based Access Control (AERBAC) model that integrates the RBAC and ABAC access control models. AERBAC retains the flexibility offered by ABAC, yet it maintains RBAC's advantages of easier administration, policy analysis and review of permissions. Moreover, we evaluate AERBAC by comparing it with RBAC and ABAC, and elaborating the features offered by each of these models. Our model has the following key features: (a) it allows context-aware access control decisions by associating conditions with permissions that are used to verify whether the required contextual information holds when a decision is made, (b) it offers a content-dependent authorization system while keeping the approach role-oriented, in order to retain the advantages offered by RBAC. This work resulted in the following publications accepted at peer-reviewed conferences:

- [117]: Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Integrating attributes into role-based access control. In *29$^{th}$ Data and Applications Security and Privacy Conference (DBSec)*, 2015, Springer, pp. 242–249. Published.
- [116]: Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Attributes enhanced role-based access control model. In *12$^{th}$ International Conference on Trust, Privacy and Security in Digital Business (TrustBus)*, 2015, Springer, pp. 3–17. Published.

4. Based on AERBAC, we develop a Role Oriented Access control Mechanism for Video Surveillance (ROAMVS). We extend AERBAC with spatio-temporal constraints and define how users and objects in video surveillance systems may be specified using the proposed model. In order to enable multilevel access control that reveals different information to different users in the same video, we define privilege modes by combining video properties with actions. We describe derivation of permissions in ROAMVS from explicitly stated permissions, due to existence of attribute hierarchies. A prototype implementation of our access control mechanism using eXtensible Access Control Markup Language (XACML) is developed to demonstrate the feasibility of the proposed access control model in video surveillance applications. The ROAMVS model offers the following novel features: (a) it provides a metadata-based yet role-oriented access control mechanism that allows to review the permissions assigned to a user, (b) it offers multilevel access control without using negative authorizations and hence avoiding any conflicts in the policy. Based on this work, a manuscript is currently under preparation which will be submitted in the IET Information Security journal:

- [115]: Rajpoot, Q. M., and Jensen, C. D.: Role-oriented Access Control

Model for Video Surveillance Systems. *Elsevier Computers & Security* (To be submitted).

## 1.5 Thesis Overview

Chapter 2 provides an overview of video surveillance systems and the access control paradigms. It highlights the privacy issues in video surveillance and describes the recent research trends to combine attributes and roles. This chapter sets the background in order to help reader understand the challenges relevant to privacy, security and access control in video surveillance, discussed in next chapters.

Chapter 3 presents an abstract model to identify the security and privacy requirements in video surveillance. This chapter reviews the existing solutions for the identified requirements and points out the challenges to be addressed. It also identifies the features desired in an access control model suitable for video surveillance systems.

In Chapter 4, we present our Attributes Enhanced Role-Based Access Control model and its formal specification. We then evaluate AERBAC by comparing it with RBAC and ABAC, and elaborating each of these models with respect to the features required by video surveillance systems.

Chapter 5 presents our access control mechanism for video surveillance and describes how the policy may be specified using the characteristics of user, objects and circumstances. It explains permission derivation due to attribute hierarchies, discusses the prototype implementation and compares our solution with other relevant approaches.

Chapter 6 summarizes the presented research and identifies future work directions.

Chapter 2

# Background

This chapter provides an overview of surveillance in its different forms while focusing in particular on video surveillance systems. We discuss a simplified architecture of video surveillance systems and its capabilities in order to help a reader understand the security and privacy issues in video surveillance systems. A major contribution of this thesis is to provide an access control mechanism suitable for video surveillance systems. Thus, towards the end of the chapter, we provide an overview of the two main access control paradigms that are most commonly used in industry and academia. We briefly discuss the advantages and disadvantages of each of these paradigms and describe the recent research trends to combine attributes and roles.

## 2.1 Surveillance

Surveillance is the act of watching the activities of people, with or without the consent of the people being watched, typically for management or security reasons. The recent technological developments have reduced hardware costs and increased the levels of automation, so governments and law enforcement agencies worldwide consider surveillance a cost-effective method for fighting serious threats to public safety.

Surveillance is increasingly used in developed countries and the majority of people are unaware of the magnitude of its occurrence in the form of our images recoded by surveillance cameras in public places, interception of our communication over the Internet, or our voices recoded during phone conversations [105]. There are several forms of surveillance and a significant amount of work in surveillance has been carried out through biometrics [84] and 'dataveillance' such as communication monitoring [86]. Several sociologists have discussed the reasons motivating the high level of surveillance experienced in modern societies along with its implications. Surveillance is viewed as a key tool of social classification, power and disciplinary control in the modern state [85]. The term Panopticon is often used to indicate the ultimate power offered by massive surveillance [53, 156]. The term Panopticon is originally coined by the English philosopher and social theorist Jeremy Bentham in the late 18th century to describe a type of building where a single watchman can observe all people from a central location [45, 129]. The Panopticon was promoted as the ideal architecture for a prison, because the fact that prisoners cannot know when they are being watched means that they always have to act as if they are currently under observation, thus effectively controlling their own behavior at all times (this is called an 'unequal gaze' by Foucault [53]). The 'unequal gaze' achieved through the Panopticon causes the internalization of disciplinary individuality, and creates the docile body required of the prisoners. This means one is less likely to break rules or laws if they believe they are being watched, even if they are not currently under observation.

### 2.1.1 Forms of Surveillance

Although, our particular focus is on video surveillance, we do provide a brief overview of other forms of surveillance before we start discussing video surveillance. The aim is to provide an overall picture of surveillance in its different forms and to give a clear idea to a reader about: (i) the scale of surveillance and the extent to which the data about activities of people is being collected both by law enforcement agencies and private organizations, e.g. Google and Facebook, (ii) the use of automation in surveillance and the general perception of people about surveillance, and (iii) to distinguish between different forms of surveillance though there exist some commonalities in terms of privacy issues and the legal infrastructure applicable to these different forms of surveillance.

Surveillance systems are normally divided into two major types based on the means using which they are conducted: i) Electronic Surveillance and ii) Non-electronic Surveillance. The former includes computer surveillance, mobile phone surveillance, workplace surveillance and video surveillance. Non-electronic surveillance, on the other hand, does not involve digital technology but makes use of

human-beings such as appointing human operatives to shadow a target or intercepting postal messages. The benefit of electronic surveillance is that it facilitates automation, e.g. by computers, so that mass surveillance can be achieved with relatively few human resources. The costs of surveillance technology (hardware and network connectivity) are decreasing, so human resources have become the most costly component of a surveillance system. The automation of surveillance systems therefore allows more people to be monitored at equal costs or an overall reduction of costs in the surveillance system. Therefore, the advent of digital technology has significantly reduced the use of non-electronic surveillance.

#### 2.1.1.1 Computer Surveillance

Computer surveillance is the act of monitoring the computer activity, data stored in the computer and the data transmitted over the network. Computer surveillance techniques that focus on the activities and data stored on individual computers are typically referred to as host-based techniques, while techniques that primarily monitor the data transmissions and traffic flows on the network are known as network-based techniques. Regardless of the techniques that are being used, computer surveillance can be either voluntary and participatory, such as the use of cookies by web-browsers, or involuntary and even surreptitious, such as the use of device fingerprinting techniques [94, 159] or the extensive amounts of log data that have been stored by European network providers since the introduction of the European Data Retention Directive [1].

Host-based surveillance techniques normally require software to be installed on the individual host. This software may be installed either by the users or system administrators for explicit monitoring purposes, as is the case with anti-virus software and other softwares installed to detect or prevent the presence of malicious software (aka. malware) on the host computer. This software typically monitors activities, e.g. running processes and subsystems, and/or data on the system, e.g. important system files, such as the content of configuration files, system and application log-files, or important data files created by the users. Some information stored on the computers may be public in nature, but should still be restricted to a limited set of authorized users. This is particularly true for online social media, such as Facebook, Linked-In, Instagram and Snapchat, which can be analyzed to extract information about a person's interests, associations, beliefs, plans and activities [7].

Network-based computer surveillance requires access to the communication infrastructure at some point between the two communicating parties. For example the Communication Assistance for Law Enforcement Act in the United States,

authorizes the law enforcement agencies to tap phone conversations and to intercept Internet traffic including reading of emails. This act requires the Internet Service Providers to install sniffing technology allowing law enforcement agencies to monitor the Internet traffic. The use of such surveillance techniques by the intelligence community have recently received much attention after the revelations of Edward Snowden, a former employee of Booz Allen Hamilton contracted to work for the National Security Agency (NSA). Edward Snowden has leaked documents that shows that the NSA is collaborating with a number of U.S. federal agencies and foreign intelligence agencies to filter Internet traffic passing through these countries [20, 142]. The implications of such mass surveillance of Internet communication is similar to the effects of video surveillance in public places that we mentioned above, i.e. people are likely to apply self-censorship and refrain from expressing opinions and views that may be considered "dangerous" by the intelligence services. As the work of the intelligence services is necessarily secret, self-censoring citizens must leave a wide margin of error, which severely limits the expression of free speech through the Internet. As with modern video surveillance, this mass surveillance capability stems from the automation made possible by computers.

Many online services offered nowadays, e.g. social media, email, data storage, are provided free of charge, or at a very low cost, which entitles the service provider to impose their own conditions on the use of their services. For instance, Google's privacy policy states that Google scans the contents of emails exchanged over its email service, Gmail, collects information about their users' Internet surfing habits and modifies cookies on their users' computers. This information is primarily collected by Google to profile their users and make their online marketing more effective, but U.S. law enforcement agencies publicly admit to using such data collected from such organizations in order to strengthen the profile of an individual under surveillance. It is common to find laws that authorize security agencies to monitor activities of their people over the Internet, in other parts of the world such as the European Data Retention Directive [1].

#### 2.1.1.2 Workplace Surveillance

Frequent usage of Internet and email at work and sophisticated computer technology allow the employers to regularly monitor the actions of their employees while they are at work. Activity logs from enterprise resource planning (ERP) systems, or customer relationship management (CRM) systems provide management with an accurate record of what, where and when their employees work. While these activity logs are typically required to comply with corporate governance legislation, they may also be abused to closely monitor the activities of employees in the workplace. Employers are continuously increasing the

monitoring spanning from monitoring of email, web surfing to tapping of office phones to enhance the productivity of the organization. This has become so prevalent that the U.S. government has published a brief discussion of what practices are legal [143]. Secondary use of data, i.e. use of data for a purpose different from the one for which it was collected, is generally not allowed, but it is difficult to prevent or detect. According to a survey conducted by American Management Association [9], more than 75% of US organizations monitor email messages, Internet usage, phone calls and computer files of their employees. More than 25% of the fired workers were dismissed for misusing of email while around 33% have been fired for misuse of the Internet. Misuses include violation of company policy, inappropriate content and excessive personal use. There are many genuine reasons for organizations to know what is happening within the organization, however, the employer is expected to remain aware of the employee's right to privacy. In most countries, privacy or data protection legislation, at a minimum, requires that employers obtain consent from the employees by stating how the organization is monitoring them, what information is being collected, the purpose of the information collection and who may review the information. Compliance with this legislation generally also prevents the employer from secondary use as mentioned above.

After a brief introduction of different forms of surveillance, we now shift our focus to video surveillance.

## 2.2 Video Surveillance

Video surveillance is a system that employs, normally, a network of cameras to monitor a particular area (public or private) for protection against theft, violence, terrorism or other similar issues. A simple system would allow a watchman to observe what is going on in an area under surveillance while a sophisticated one may include thousands of cameras linked together making use of state-of-the-art technology, e.g. object-detection, rapid data retrieval (cf. § 2.2.1).

Video surveillance systems are of different types depending mainly on the area where they are deployed, e.g. publicly accessed area, private area. The type of video surveillance systems determines the scale of such a system, the security and privacy issues, as well as the legislation applicable in these systems. In this context, Fig. 2.1 offers a taxonomy of video surveillance systems as a hierarchical structure. We classify video surveillance into two major types: i) Video surveillance in publicly accessible areas, and ii) Video surveillance in private areas. Publicly accessible areas such as streets, public transport, shopping malls, supermarkets, etc. are accessible to anyone and anything observed here may

be observed by everybody. On the other hand, private areas such as offices are accessible to a limited number of people whose identity may already be known. Video surveillance in publicly accessible areas can be performed either by public authorities (e.g. law enforcement authority) or by private sector (e.g. owner of a supermarket) and each has to follow the relevant laws. Video surveillance in private areas can also be categorized into two subcategories: i) imposed by third party, and ii) self-initiated. In the former, the video surveillance is initiated by a third party with/without consent of the people under surveillance, example includes workplace video surveillance; while in the latter category, it is initiated by the people under surveillance themselves and example includes use of video surveillance in one's own home.

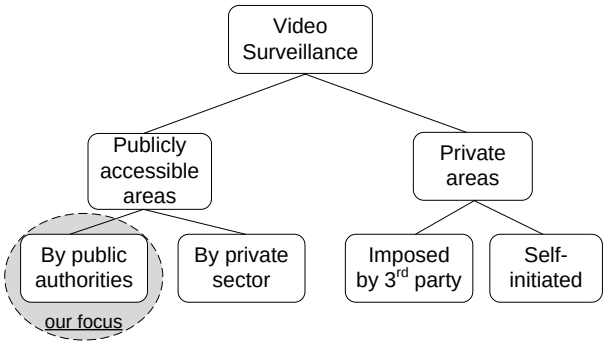


**Figure 2.1:** Taxonomy of video surveillance system

In this thesis, our focus, in particular, is to address security and privacy issues in video surveillance held in publicly accessible areas (also known as CCTV) which are accessed by public authorities such as city/council administration staff, police, etc.

The potential benefits of CCTV especially with respect to security are seen as a cost effective mechanism to fight severe threats to public safety. People are monitored in public areas like train stations, buses, stores and around ATMs sometimes without even noticing. In UK alone, there exist more than 4 million cameras [97] and a report from BBC News [18] estimated that an average person in London is caught on camera around 300 times a day. The pervasive form of video surveillance systems combined with the technological advances have the potential to disrupt the balance between the need for such systems and the privacy of individuals [140]. To better understand the privacy concerns raised due to usage of video surveillance, the issues related to security and privacy in video surveillance and the importance of access control; below we discuss a simplified architecture of a modern video surveillance system and the capabilities

of such systems.

### 2.2.1 Capabilities of Video Surveillance Systems

In contrast to the early age and many currently deployed CCTV cameras, which can only see as far as a human eye and have fixed direction, modern cameras can pan and tilt and can provide a lot more detailed image than previously possible. A camera having a 60-times optical zoom lens can read what is written on a cigarette pack at 100 yards [132]. Furthermore, in a report by New York Times [93], 400-times magnification cameras have been deployed in Chicago. Improved quality of recordings, reduced storage costs and use of digital technology enable traversing and exploitation of recorded data in ways previously impossible with analog recordings.

The use of advanced video analytics techniques such as object-detection and event-detection is continuously increasing in modern video surveillance systems [58]. Compared to these modern solutions, traditional CCTV systems are simply monitored by human observers without automated technological assistance. The modern systems facilitate rapid data retrieval and make it easy to search for a particular person or activity and may lead to profiling of individuals [134]. These techniques may allow those monitoring the systems to perform voyeurism and gather unauthorized data about activities of an individual [134]. Doing so requires a significant amount of time and effort in traditional surveillance systems, however, there still have been reported incidents of voyeurism (cf. § 2.3.3) in traditional systems. Therefore, the privacy concerns in modern video surveillance systems become obviously much more serious.

Although facial recognition and other remote biometric systems [64] are yet in their infancy, there is a significant investment in this area and the reliability of the identification process is improving [140]. Advancements in this area can be integrated with video surveillance systems to track movement in their field of view or across networked cameras allowing those monitoring the system to automatically follow a target object in an entire city [87]. This means that people can be tracked in real time and with little effort, which makes cyber stalking quite easy for anyone with access to the surveillance system, regardless of whether this access is authorized or not. People participating in political rallies can be followed to their home address as the meeting dissolves and any people visiting celebrities or political dissidents can be followed on camera by paparazzi or law enforcement agents in oppressive regimes. Due to the advent of modern video surveillance systems that facilitate rapid data retrieval enabled by searching and advanced imaging technology, massive usage of CCTV in public places is of great concern for civil libertarians and is seen as a threat to privacy

by critics [76] (cf. § 2.3).

### 2.2.2 Architecture of a Video Surveillance System

Modern video surveillance systems primarily use the Internet as the medium to transfer data to intermediary servers, storage systems and the users. Such a system normally employs a network of several cameras which capture video data at their respective locations, as depicted in Fig. 2.2. This data is sent to the storage server responsible for securely storing the data. Depending on the application requirements, this could be a centralized or distributed storage solution. The data may be accessed by users, wishing to see the live or recorded data of a desired location, e.g. live video feeds are often sent to a special monitoring room, and this live or stored data may also be watched on hand-held devices or a workstation. As stated earlier, we refer to such users as observers. In order to authorize access of the recorded videos, the control unit handles access requests from the observers and allows them to access data as per the specified policy.

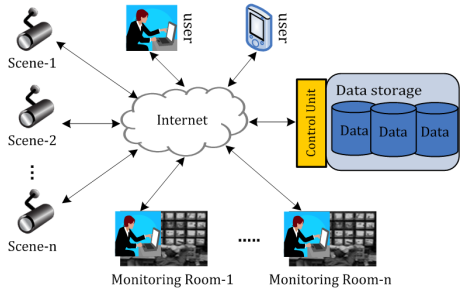


**Figure 2.2:** Architecture of a video surveillance system

#### 2.2.2.1 Camera and Video Characteristics

Each camera in the surveilled area is associated with a location where it is deployed. The location coordinates (physical location) are mapped to semantic division of locations (logical location), suitable for a given application, e.g. for a video surveillance system covering New York City, the five boroughs, Manhattan, Brooklyn, Queens, Bronx and Staten Island, could define a coarse grained division of logical locations. These regions could be further divided into subregions corresponding to the neighborhoods that constitute each borough. Depending upon the type of camera used, a camera orientation may be moved into

different directions, e.g. left, right, where each direction provides a different field of view and hence records different information to be viewed. Once recorded, a video may be shown with different quality and information-levels to different users in different circumstances. For example, the resolution of the video may be changed to lower resolution while hiding the privacy sensitive regions, e.g., faces, such that the behavioral information of people recorded in the video is shown to the observer but their identity is not revealed.

The information contained in the video may be extracted in the form of annotations, using the previously mentioned video analytics techniques. The annotations extracted from the video may include the objects (e.g. humans, vehicles) and events (e.g. fire, burglary) contained in the video. These annotations and other information like the location and time of video recording are part of the video metadata. This metadata is stored linked to the videos in order to allow retrieval of videos based on metadata information. In order to allow content-based searching and retrieval of videos, longer videos are normally segmented into smaller video units, called video shots, and each video shot is associated with the metadata [79].

#### 2.2.2.2 Example System

In order to show the importance of current situation (circumstances) in controlling access to video surveillance data and to clarify the difference between static and dynamic access control, we discuss an example system below.

Consider the video surveillance system deployed in the region of Manhattan, for instance. This system consists of several cameras which are deployed in the major public places all over the region. The captured data is continuously monitored manually, along with the technological assistance by the system which generates an alarm upon detection of an anomalous event e.g. crossing a monitored fence. The observers are associated with different areas of the Manhattan region, e.g. Manhattan_south, Manhattan_north, and on generation of an alarm they investigate closely what happened and send a patrol or call the police, if required. The observers may access the data in the monitoring room or on their hand-held devices when they are approaching the place of incident. However, notice that the observers are normally pre-associated with the specific areas and are already granted access to watch videos of those areas, independent of the alarm generation. This is called static access control where observers are always allowed to access the data, independent of the current circumstances (e.g. alarm, location of observer). Such an access control leads to privacy issues and increases the chances of performing voyeurism by the observers.

#### 2.2.2.3 Dynamic Access Control

An alternative approach could make use of dynamic access control where access to the data is granted to the nearest available mobile observers only upon detection of an event. Considering the proportionate access principle, observers in the monitoring room may be given regular access with less privileges (e.g. low resolution) in normal situations and higher privileges in an emergency situation.

Using these techniques can prove to be immensely useful in public video surveillance. Allowing access of data to certain individuals only in case a specific event occurs or in an emergency situation, addresses the privacy concerns raised because of continuous video surveillance. Suppose there is a fire incident reported near Times Square. Upon detection of this emergency situation, along with the observers assigned to this location, the nearest fire-brigade and police stations are also informed about the event and the system allows access to video data to the respective employees of these stations. Allowing access to the video stream to the fire-brigade and police station would help them understand the severity of the situation and to come prepared with appropriate tools and man-power to better combat such situations. Although the system should allow advanced functionalities such as searching, tracking an individual and automatically identifying an individual, however, appropriate access control mechanisms (cf. § 2.4) must be adopted in order to minimize the chances of performing voyeurism by the observers, reduce privacy invasion and to make these systems widely acceptable.

In the following, we discuss the importance of privacy and the concerns raised by civil libertarians regarding usage of mass video surveillance.

## 2.3 Privacy Concerns

The current and potential capabilities of video surveillance systems are quite attractive for law enforcement officials worldwide and they see video surveillance as an effective mechanism to fight against security threats. Critics, however, argue that being pervasive in nature video surveillance poses a threat to many democratic rights of the non-criminals and it might force law-abiding people to change their daily routines in order to avoid being caught by the camera. Few privacy awareness initiatives like the Isee project in Manhattan [10] and the Observing Surveillance Project in Washigton [103] identified locations of CCTV cameras to help people avoid being captured by the cameras. Advent of technology with its capacity to collect and analyze information about individuals

increased interest in the right of privacy. Below we discuss the privacy concerns raised by the extensive use of video surveillance.

### 2.3.1 Threats to Privacy

Though there does not exist a universal definition of privacy, it is often described as how far society can intrude into personal affairs of an individual. A well-known definition of privacy given by Alan Westin [150], author of "Privacy and Freedom", is: "the desire of people to choose freely under what circumstances and to what extent they will expose themselves, their attitude and their behavior to others". In many privacy theories it is considered an individual right. For example Thomas Emerson [47] states that privacy is "based upon premises of individualism, that the society exists to promote the worth and the dignity of the individual... The right of privacy... is essentially the right not to participate in the collective life — the right to shut out the community".

Presently, most of the countries in the world recognize the right of privacy in their legislation. In some countries, for example the United States, where the right of privacy is not explicitly mentioned in the constitution, the courts have ruled inferring this right from other provisions. Existing legislations to protect the privacy of citizens in the context of lawful video surveillance vary greatly among different countries with respect to the different aspects of video surveillance that are being regulated. For example, there are different rules for limiting the storage time of recorded images, the need for notification signs in the surveillance area and the possible requirement of a court warrant in order to perform surveillance on a particular person. For a classification of legislation relevant to video surveillance in different countries and the guidelines to achieve legal compliance, see Appendix-A.

The boundary between what we reveal and what we do not and control over that boundary, are among the most important attributes of civilization [89]. Many people do not feel comfortable in exposing their behaviors to strangers even if they do not fear disapproval or hostility by the society. Thus there are certain democratic rights which are at stake because of extensive video surveillance. Firstly, the right of people to freely express their thoughts and to associate freely to share those thoughts is in danger. People might not feel comfortable to express their views or to take part in protests against the government policies if they knew they might be identified for this activity later on. Second such right at stake is the right of anonymity which is closely related to privacy. Many people expect to remain anonymous in public places such as entering an infertility clinic or a psychiatrist's office. The presence of video cameras in public places would capture all such activities and would allow the officials to see daily activities of

any individual.

### 2.3.2 Counter Arguments

#### Nothing to Hide, Nothing to Worry About

A typical argument that is often presented in discussions about privacy issues is: “If you have got nothing to hide, you have got nothing to worry about” [147]. A similar argument was presented as a slogan by the British government in a campaign to support video surveillance [121]. Frequently encountering such an argument in news interviews and discussions, Daniel Solove [135] states that he decided to ask the readers of his blog to provide their opinions in response to this argument. Some interesting comments he received in response include:

- This is not about hiding something, this is about it being none of other people’s business
- I am doing nothing wrong and do not need to justify my position. If you need to investigate my activities, get a warrant to do so
- I do not have anything to hide, but I do not have anything I feel like showing you, either

The reasoning of the argument nothing to hide depends on the fact that privacy is violated only if something illegal or embarrassing is revealed about an individual. Hence the majority of people not involved in such activities has nothing to worry about. Rephrasing the argument in a generic manner that "*all law-abiding citizens should have nothing to hide*" reveals that nothing to hide argument is misleading and is based on a wrong assumption that privacy is about hiding wrong-doings. Concealment of bad things is just one aspect of privacy among many other aspects like lack of transparency and accountability and usage of collected data for purposes other than the informed ones. The nothing to hide argument attempts to hide the existence of a problem altogether [136].

#### No Privacy at Public Places

In a similar context, another point which is frequently debated is, when you are in a public space such as in a mall or a street, every step you take may be watched by someone anyway so what difference does it make whether you are watched by a person or a camera. The fundamental difference is symmetry. When you are being watched by a person you can watch them back however, when you are being watched but cannot watch them back forms an asymmetrical

relationship. Consider a one-sided mirror between an employee and employer's room to understand the situation. Thus the core of the debate is not the fact that whatever you do may be watched by someone rather the opposite that there may be a particular person who is watching everything you do, facilitated by the automated large-scale video surveillance without much effort and cost [147].

### 2.3.3 Misuse of Video Surveillance

A potential threat in video surveillance systems is voyeurism – exploitation of video surveillance system by the authorized personnel for targeted collection of data on activities or behaviors of an individual [96]. According to a report by BBC News [19] a few council workers in Liverpool spied on a woman's apartment using a modern pan-tilt-zoom CCTV street camera. Such misuse can be extended to spy on government officials or celebrities. For example, in another incident which started a whole new debate about use of CCTV, a security guard used a museum's CCTV camera to spy on the German Chancellor Angela Merkel's private apartment [34].

Another potential problem of video surveillance system is its discriminatory use by the officials against a particular individual or community based on the ethnic, racial, gender or religious grounds. For instance, Norris & Armstrong [95] found in their study about CCTV surveillance in the UK that black people are twice as likely to be a target of surveillance as compared to white people and similarly men are three times more likely to be surveilled than women, not because of their involvement in crime or disorder but simply based on categorical suspicion.

Despite posing a threat to privacy and dangers of its misuse by the officials, the usefulness of video surveillance systems cannot be denied. What is mainly needed is that these systems must be designed in ways that not only protect privacy and freedom, while protecting the people against security threats, but they must also be able to prevent or detect any abusive usages by using techniques such as logging, encryption and access control mechanism.

## 2.4 Access Control

As mentioned earlier, we focus on addressing the security and privacy in video surveillance through the use of dynamic access control mechanism. This section describes the most influential access control paradigms and the current focus of research in the area of access control.

Access control models are of critical interest in computer security and have been used as an important protection mechanism since the advent of multi-user computing. The access control system determines whether or not a request to access a resource is to be granted. The requesting entity is typically referred to as a subject (a program or process representing a user) whereas the resources to be accessed are called objects. The terms user and subject are used alternatively in this thesis.

Since the introduction of the access control matrix in the late 1960's, several access control models have been proposed. Out of these, the most successful in terms of practice are: Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-based Access Control (RBAC) and Attribute Based Access Control (ABAC). DAC allows the creator (owner) of an object to make decisions about who is to be given what permissions on that object. In order to have a centralized access control system, managed by the system owner rather than the object owner, MAC allows to apply further constraints that are beyond the control of object owner, e.g. by classifying both subjects and objects into different security levels. MAC enforces constraints on the information flow and addresses the problem of unauthorized access on copies of objects, faced by DAC. A subjects request is granted only if the object to be accessed belongs to a specific class which is accessible by the security level assigned to the subject. In contrast to MAC which targets military settings, RBAC addresses primarily the access control requirements of commercial organizations. RBAC was introduced in early 90's by Ferraiolo et al. [43] to address the problems faced by the then dominant DAC and MAC models. Sandhu et al. [124] later proposed a framework of RBAC models, *RBAC96*, categorizing RBAC into conceptual models. After further modifications, RBAC was approved as NIST standard ANSI/INCITS 359-2004 [50]. While RBAC offers several advantages, it has certain limitations too (discussed below). A relatively new access control paradigm, Attribute-Based Access Control (ABAC) [161] has been identified to overcome these limitations of RBAC [38].

Below we describe the currently dominant RBAC and ABAC access control paradigms. We also discuss few solutions using these approaches and the issues with these approaches followed by the need to combine RBAC and ABAC.

### 2.4.1 Role-Based Access Control

RBAC is the most popular access control model and has been a focus of research since last two decades. In RBAC, the roles encapsulate permissions and users are assigned to roles in order to obtain a permission, as shown in Fig. 2.3. The RBAC model [50] comprises of the following four components: USERS,

ROLES, OPS, OBS and PRMS representing the sets of users, roles, operations, objects and permissions, respectively. A user is a human being represented by a program or a process. A role represents a certain job function, within an organization, associated with a set of tasks that the role is entitled to perform. An operation allows a user to execute a function, e.g. read, write, delete, on objects. An object is an entity that represents information or a system resource. Examples of an object include files, database tables, printers, CPU cycles etc. A permission is an authority to perform an operation on objects. Each permission defines the operation, an element of the set OPS, that can be performed on a particular object, an element of the set OBS. A session relates a user to a subset of roles assigned to the user. The natural notion of role allows using role hierarchies and it also supports the prevention of the conflict of interest by specifying the separation of duty (SoD) constraints. Static separation of duty (SSD) constraints are defined on user-role assignment and role hierarchies whereas dynamic separation of duty (DSD) constraints are defined on the roles which may be activated by a user during a session. Once a user is authenticated, the user can request to create a session by activating a set of roles that the user is entitled for. In case the activation request is approved, the user obtains all the permissions relevant to the activated roles.

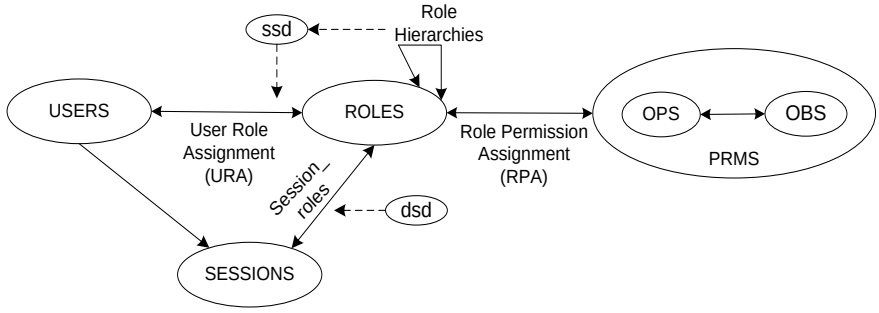


**Figure 2.3:** NIST Role-Based Access Control [50]

Table 2.1 provides the sets and functions used in RBAC. A fundamental concept in RBAC is to assign the users to the roles and the permissions are assigned to roles. This provides a way to assign multiple permissions to a user by simply assigning a role to the user. A user can be assigned to one or more roles, and a role can be assigned to one or more users. This many-to-many relationship between users and roles is represented by URA. Similarly, relationship between roles and permissions is represented by RPA. A session can be associated with a single user whereas a user can create one or more sessions. The function *user_ sessions* returns the set of sessions associated with a given user. The function *session_* -

*roles* returns the roles activated in that session and the function *session_ user* gives us the unique user to whom a given session belongs.

**Table 2.1:** Sets and functions used in RBAC

- *USERS, ROLES, OBS,* and *OPS* (users, roles, objects and operations respectively).
- $URA \subseteq USERS \times ROLES$, a many-to-many mapping of user-to-role assignment.
- *assigned_ users*: $(r{:}ROLES) \rightarrow 2^{USERS}$, the mapping of role $r$ onto a set of users. Formally: $assigned\_users(r) = \{ u \in USERS \mid (u, r) \in URA\}$.
- $PRMS = 2^{(OPS \times OBS)}$, the set of permissions.
- $RPA \subseteq PRMS \times ROLES$, a many-to-many mapping of permission-to-role assignment.
- *assigned_ permissions*$(r{:}\ ROLES) \rightarrow 2^{PRMS}$, the mapping of role $r$ onto a set of permissions. Formally: $assigned\_permissions(r) = \{ p \in PRMS \mid (p, r) \in RPA\}$.
- *SESSIONS*, the set of sessions.
- *user_ sessions*$(u{:}\ USERS) \rightarrow 2^{SESSIONS}$, the mapping of user $u$ onto a set of sessions.
- *session_ roles*$(se{:}\ SESSIONS) \rightarrow 2^{ROLES}$, the mapping of session $se$ onto a set of roles. Formally: $session\_roles\ (se_i) \subseteq \{ r \in ROLES \mid (session\_user(se_i), r) \in URA\}$.
- *avail_ session_ perms*$(se{:}\ SESSIONS) \rightarrow 2^{PRMS}$, the permissions available to a user in a session.

#### 2.4.1.1 RBAC Advantages

RBAC is simple to manage and facilitates reviewing permissions assigned to a user, due to encapsulating privileges into roles [54]. It makes the task of policy administration less cumbersome, as every change in a role is immediately reflected on the permissions available to users assigned to that role [74]. The role-based approach fits into organizational structure quite naturally because the employees in organizations typically assume job functions which are associated with well-defined responsibilities and tasks [63]. According to a report by National Institute of Standards and Technology (NIST) [104], the adoption of RBAC in commercial and government organizations is continuously increasing.

As discussed earlier, we need to consider the current circumstances, also called context, e.g. occurrence of an event, generation of an alarm, user's current location etc., when a user's access request is to be evaluated in video surveillance systems. Below we discuss how context is handled in RBAC and the issues faced.

#### 2.4.1.2 Context in RBAC

Due to the advent of pervasive systems, access control has become more complex as access decisions may depend on the context in which access requests are made. The contextual information represents a measurable primitive and may entail such information being associated with a user, object and environment [37]. For example, an access control policy may depend on the user's *current location*, the object being currently in a *specific state*, and the *time of day* when the access is requested. Such information is captured via user attributes, object attributes and environment attributes (cf. § 4.1). However, standard RBAC does not take into account the situations where contextual attributes are required parameters in granting access to a user. Several efforts have been reported which extend RBAC to include contextual information such as location, time, system load, etc. Few of the key works are mentioned below.

The solution presented by Moyer et al. [88] extends traditional RBAC by presenting novel concept of using environment and object roles in addition to subject roles in RBAC. The motivation behind using the notion of role is to exploit features associated with roles such as role-hierarchy and separation of duties. Zhang et al. [162] present context-aware access control mechanism for pervasive computing environments. The solution extends RBAC to make use of environmental information and introduces context agents which are associated with each subject and resource in order to keep track of contextual information. Kim et al. [71] extend RBAC targeting ubiquitous computing where the roles of the users are adjusted considering the contextual information. A state checking agent collects users context and verifies it against state checking matrix (SCM) to determine if the assigned roles should be active or deactive. Bertino et al. [22] attempt to formulate a framework which can be used to model geographical (spatial) information. It extends RBAC to deal with location information while granting access. The users are assigned geographically bounded roles which are activated if user's location lies in the restricted boundary. A model to express temporal constraints on enabling of roles and user-role assignment has also been proposed [21]. Ray et al. [119] propose a model that extends RBAC with spatio-temporal information. The model enables enforcing spatio-temporal constraints on role-activation and permissions. Solution proposed by Kulkarni et al. [75] extends RBAC by combining the already existing approaches including context based user to role assignment [68], context based permission assignment [92] and applying context when evaluating access request [118]. Filho et al. [29] extend RBAC for medical information systems and propose a solution which considers contextual information not only related to a user but also related to the owner of a resource (patient). Such a setting assigns utmost value to the owner's priority and allows, for instance, a patient to specify that any doctor may access his information in emergency cases. Several other solutions extending RBAC with

contextual information have also been proposed [36, 60, 68, 120, 5].

The above-mentioned solutions, however, typically require creation of a large number of closely related roles, causing a role-explosion problem. Moreover, a fundamental limitation of these approaches, inherited by RBAC, is that the permissions are specified in terms of object identifiers, referring to individual objects. In situations where a user may have similar access rights to large number of objects, e.g. in the hundreds of thousands, a permission must be created for each object, which leads to a permission-explosion problem. Solutions to address the issues of role-explosion and permission-explosion have also been proposed [55, 57, 69] which we discuss in § 4.4.

### 2.4.2 Attribute-Based Access Control

During the last few years, ABAC [161, 42] has been emerging as an alternative approach to RBAC. It is argued that ABAC has the capability to overcome the limitations of RBAC, such as incorporation of contextual information in making of access control decisions [74]. An attribute is a name:value pair that can represent just any information relevant to an entity including user, object and environment. The access control policy in ABAC is formed using rules which are constructed using the attributes of users, objects and environment. For a user request to be granted, there must exist a rule in the access control policy that authorizes a user to access an object by specifying the attributes associated with the user, object and environment. Although the ABAC model was first introduced by Yuan et al. [161], the use of attributes for users and objects has been recognized in the literature since the inception of distributed and internet-based applications (e.g.,[26, 155, 24, 4]).

A reasonable amount of research has been carried out following ABAC approach. Realizing the well-known problem of role-explosion in RBAC, Yuan et al. [161] argue that service oriented architecture (SOA) in web services environment requires a dynamic and fine-grained access control model. Damiani et al. [42] proposed an attribute-based framework for open environments. Hai-bo et al. [131] discuss that attribute-based approach is well-suited for web-services because of its distributed and dynamic nature. Covington et al. [37] propose to use contextual attributes for authorizing access without relating them with an identity or role. Park et al. [108] divided attributes into two categories: non-mutable and mutable. They developed a mechanism where values of mutable attributes may be updated based on operations performed by the user. In a quest to propose a formal reference model for ABAC, similar to the one that exists for RBAC, Jin et al. [66] worked on defining the core components in an ABAC model in order to provide foundations for a widely agreed-upon reference

ABAC model.

ABAC approach is considered more flexible as compared to RBAC, since it can easily accommodate contextual attributes as access control parameters [74]. However, the administrative advantages of RBAC such as reviewing the permissions associated with a user are lost as ABAC is typically much more complex than RBAC in terms of policy review. Thus analyzing the policy and reviewing or changing user permissions are quite cumbersome tasks in ABAC [38]. We discuss the features and limitations of RBAC and ABAC in § 4.3.2, in more detail.

### 2.4.3 Combining RBAC and ABAC

On one hand, both RBAC and ABAC have their particular advantages and disadvantages. On the other hand, both have features complimentary to each other. This fact led to announcement of NIST initiative [74] to integrate RBAC and its various extensions with ABAC in order to combine the advantages offered by both RBAC and ABAC.

The NIST initiative identified three possible ways in which roles and attributes may be combined. The first option is *dynamic roles*, where attributes determine the roles to be activated for a user. Al-Kahtani et al.[6] and Kern et al.[70] explored this option for automated user-role assignment, in large organizations, using attribute-based rules. These solutions consider only user attributes and do not address the issues of role-explosion and permission-explosion. In the second approach, roles and attributes may be combined in an *attribute-centric* manner. In this approach, the roles are not associated to permissions; rather they are treated as just one of many attributes. This approach is essentially the same as ABAC and does not inherit any benefit from RBAC. In the third approach, called *role-centric*, roles determine the maximum permissions available to a user, and attributes are used to constrain these permissions. Kuhn et al.[74] identify this approach as a direction for future research, since it may retain the advantages of RBAC while adding the much needed flexibility.

Our AERBAC model combines roles and attributes using the role-centric approach. We discuss AERBAC model and few other solutions responding to the NIST initiative in Chapter 4. Later, we define an access control mechnism for video surveillance systems in Chapter 5, based on the AERBAC model. The existing access control approaches proposed for video surveillance systems and other relevant domains e.g. multimedia applications, are discussed in § 3.3.3 and § 5.4 where we also compare then with our approach.

## 2.5 Summary

This chapter discussed different forms of surveillance and a taxonomy of video surveillance systems. The capabilities of modern video surveillance systems and the privacy concerns due to usage of video surveillance systems were also discussed. We also described the major access control paradigms and the advantages and limitations of the currently dominant RBAC and ABAC access control models. Several extensions of RBAC to overcome its limitations and some research efforts using ABAC approach were also reported. Finally the need to integrate RBAC and ABAC, as initiated by NIST, and different strategies to combine these two approaches were discussed.

Chapter 3

# Security and Privacy: Requirements and Challenges

In this chapter, we present a general model of video surveillance to help identify a list of security and privacy requirements in a video surveillance system. We provide an overview of existing solutions proposed to fulfill the major requirements identified through our video surveillance model and point out their problems. We identify a potential gap where research efforts need to be put in by pointing out challenges that need to be considered while designing security solutions in this regard. Out of the identified challenges, we chose to work on dynamic access control in video surveillance systems. Thus we also identify features desired in an access control model suitable for video surveillance systems, towards the end of this chapter.

## 3.1 Video Surveillance Model

In this section, we generalize the architecture, presented in § 2.2.2, into an abstract model of video surveillance as a method to identify the manifold security

and privacy requirements in a video surveillance system. Fundamentally, a video surveillance system must include elements to capture video, to store/record video and to display video to the users, as well as a mechanism to transport video data between these elements. Figure 3.1(a) shows the main elements of our model, which includes four components, namely: video-capture, -transport, -monitoring, and -storage. The video-capture component includes the cameras, their local infrastructure, and the area which can be captured by the cameras. Once the data is captured, it needs to be securely transported; this is typically done over the Internet, so we have included this as a component in our model. It is important that video transport is done in a way that ensures the confidentiality and integrity of data while in-transit. The transport component considers transport of data from cameras to storage servers, between storage servers, and when transferring the live or stored video data to the observers. The monitoring component includes the different elements that are necessary to allow somebody to watch the video. The monitoring component must consider all security and privacy concerns that arise when the captured data (live or stored) is watched by the observers. It also includes any automatic or manual processing (e.g. masking the identity revealing regions) for the purpose of observing live or stored data, therefore when the stored data is watched by the observers, it falls under the monitoring component. Finally, the storage component is responsible for securely storing the data.

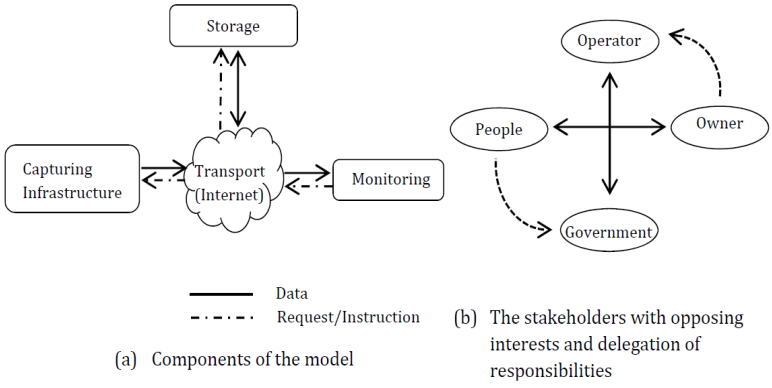


**Figure 3.1:** Video surveillance model

The four components identified in Fig. 3.1(a), allow us to identify the domain and scope for many of the security and privacy requirements that may arise in video surveillance systems. We do, however, also need to consider the different stakeholders and interests in order to identify all the security and privacy requirements in video surveillance systems. There are two principal stakeholders in a video surveillance systems, the *owner*, who commissions and is responsible for the system, and the *people* who are being watched by the system; these are

shown as principal opposing forces in Fig. 3.1(b). In practice, however, normally owners do not operate the video surveillance systems themselves, but instead delegate this task to another organization, e.g. a guard company; this organization is referred to as the *operator*. Similarly, most people are unable to determine whether video surveillance is fair and warranted or excessive, so it is typically an elected *government* which regulates video surveillance through legislation and guidelines. This means that, in practice, the video surveillance operator and the government become the real opposing forces in a video surveillance system. Note that the term observer used earlier holds a subset of responsibilities of the operator. The term observer represents a user who wishes to access the data captured in a video surveillance system, whereas operator may have additional responsibilities such as maintaining the system and ensuring legal compliance as per government regulations.

People are the core of our model, because they may have certain expectations from each component of the video surveillance system, whereas the other entities strive to live up to the expectations of the people. It is the combined responsibility of the owner and the operator to ensure the security of the system and the privacy of the people as it is defined by the government. Privacy of people should be protected both from outside attackers and the personnel within the owner and operator organizations. The operator is responsible for performing his duties while being least intrusive as far as the privacy of people is concerned.

Based on our model, requirements capturing consists of two stages. In the first stage, we map the requirements from the perspective of each of the stakeholders for each of the four components in the model. In the second stage, we remap these requirements in terms of privacy and security aspects. The first stage ensures that we identify the requirements that can be specified by the people and/or the government, owner and operator in the form of security and privacy related functionalities and features in the system.

Based on the requirements specified by the people/government, owner and operator, we then derive further requirements from the implementation point of view. For instance, the proportionate access requirement specified by the owner is divided into multiple requirements including data hiding, dynamic access control and voyeurism protection when considered in the implementation perspective. Table 3.1 presents the security and privacy requirements in video surveillance identified as a result of the first stage.

**Table 3.1:** Security and privacy requirements in different phases of video surveillance corresponding to all the stakeholders. The last column derives implementation requirements from the ones on left

| **Stakeholders/ Phase** | **People/ Government** | **Owner** | **Operator** | **Implementation requirements** |
|---|---|---|---|---|
| **Capture** | *c1.* Consent<br>*c2.* Signage<br>*c3.* Anonymity | *c4.* No data missing<br>*c5.* Availability<br>*c6.* Video capturing properties | None | *c7.* Security of software and hardware infrastructure |
| **Transport** | *t1.* Confidentiality<br>*t2.* Integrity<br>*t3.* Authenticity | *t1.* Confidentiality<br>*t2.* Integrity<br>*t3.* Authenticity | None | *t4.* Camera authentication<br>*t5.* Data encryption<br>*t6.* Key management<br>*t7.* No deletion of data |
| **Monitoring** | *m1.* Privacy safeguards<br>*m2.* Authorized access<br>*m3.* Public access to their data | *m4.* Continuous monitoring<br>*m5.* Proportionate access<br>*m6.* Occasional access<br>*m2.* Authorized access | *m7.* Data freshness<br>*m8.* Time-stamping<br>*m9.* Easy to search | *m10.* Dynamic access control<br>*m11.* Data hiding<br>*m12.* Voyeurism protection<br>*m13.* User management<br>*m8.* Time-stamping |
| **Storage** | *s1.* Secure storage | *s2.* Secure data storage as per law<br>*s3.* Deletion after retention period | None | *s3.* Deletion after retention period<br>*c7.* Security of software and hardware infrastructure<br>*t2.* Integrity<br>*t5.* Data encryption<br>*t6.* Key management |

Using our model, the first stage produces a large number of requirements. However, it contains certain overlapping and repetitive requirements too. This is because our model identifies each requirement in the perspective of the individual stakeholders. Thus in the second stage, we remap those requirements considering the conventional security and privacy aspects that allows us to combine the repetitive requirements together. Table 3.2 depicts this mapping. We briefly describe these requirements in greater details below.

***Privacy:***

*Consent and Signage: Consent (1a in Table 3.2)* of the people who can potentially be recorded by the video surveillance system needs to be obtained in advance, either explicitly or implicitly. One way to obtain consent is by informing the people about the video surveillance through *signage (1b)* i.e. displaying clear and visible symbols in the area where video surveillance takes place.

**Table 3.2:** Remapping of the requirements in Table 3.1 in terms of privacy & security aspects

| Components/ P&S Aspects | | Capture | Transport | Monitoring | Storage |
|---|---|---|---|---|---|
| **Privacy** | **1. Privacy** | *1a.* Consent (c1)<br>*1b.* Signage (c2)<br>*1c.* Anonymity (c3)<br>*1d.* Video capturing properties (c6) | None | *1e.* Privacy safeguards (m1)<br>*1f.* Data hiding (m11)<br>*1g.* Voyeurism protection (m12)<br>*1c.* Anonymity (c3) | *1g.* Voyeurism protection (m12) |
| **Security** | **2. Confidentiality** | Covered by 7a, below | *2a.* Data encryption (t5)<br>*2b.* Key management (t6) | None | *2a.* Data encryption (t5)<br>*2b.* Key management (t6) |
| | **3. Integrity** | Covered by 7a, below | *3a.* No deletion of data (t7)<br>*3b.* Integrity (t3) | *3c.* Data freshness (m7) | *3b.* Integrity (t3) |
| | **4. Authenticity** | Covered by 7a, below | *4a.* Camera authentication (t4)<br>*4b.* Time-stamping (m8) | *4b.* Time-stamping (m8) | None |
| | **5. Availability** | *5a.* No data missing (c4) | None | *5b.* Easy search (m9)<br>*5c.* Continuous monitoring (m4) | Covered by 7a, below |
| | **6. Authorized access** | None | None | *6a.* Public access to their data (m3)<br>*6b.* Occasional access (m6)<br>*6c.* Proportionate access (m5)<br>*6d.* Dynamic access control (m10)<br>*6e.* User management (m13)<br>*6f.* Logging (m14) | None |
| | **7. Others** | *7a.* Security of software and hardware infrastructure (c7) | None | None | *7a.* Security of software and hardware infrastructure (c7)<br>*7b.* Deletion after retention period (s3) |

*Anonymity, Data Hiding and Privacy Safeguards:* As the system is supposed to monitor the behavior of the people, it should strive to maintain the *anonymity (1c)* of the people by hiding their identity using certain *privacy safeguarding mechanisms (1e)*. Therefore the system must implement *data hiding (1f)* techniques which obfuscate the identity-revealing regions in the images when the observers monitor video streams in a normal situation. Needless to say, these data hiding techniques should be reversible such that identity could be revealed if required, for example while investigating a crime.
*Video capturing properties (1d):* The owner needs to determine whether cameras with advanced functionalities such as pan-tilt-zoom, night-vision and high-resolution are really required to be used, with respect to the purpose of the surveillance conducted.
*Voyeurism protection (1g):* In order to restrict voyeurism, advanced functionalities such as searching, identifying and tracking an individual are only to be made available when an observer explicitly requests them. While granting these privileges the system logs the request along with the information about the circumstances.

***Confidentiality:***
The people and owner desire that the data is accessible only to the intended recipients. *Confidentiality* ensures privacy protection against outsiders mainly when data is in transit, whereas *privacy* is a much broader concept that covers privacy protection against insiders too. *Confidentiality* can be ensured by using appropriate *data encryption algorithms (2a)* and taking care of *key management (2b)* issues. Because of the nature of the system, the encryption mechanism should be efficient enough enabling the data to reach the other end in real-time.

***Integrity:***
Any unauthorized change in the data should be detectable. Appropriate measures should be taken to ensure the *integrity (3b)* of data. Moreover, it should not be possible to *delete chunks of data (3a)* while leaving other data intact so as to hide the data captured in a specific time interval.
*Data freshness (3c):* The operator requires newly captured data in live streaming rather than previously captured data being replayed.

***Authenticity:***
*Camera authentication (4a):* In order to ensure the authenticity of the captured data, each camera may be required to authenticate itself to the server.
*Time-stamping (4b):* The recorded data must include verifiable time-stamping helping to ensure that the data was captured at a specific time and also to search videos specifying the time interval later on.

***Availability:***
The services offered by the system should of course be *available* when needed. If surveillance takes place upon detection of an event e.g. motion detection then such a mechanism is to be made perfectly reliable such that no event goes uncaptured i.e. *data missing should not be possible (5a).*
*Easy search (5b):* The operators require that advanced functionalities such as searching, identifying and tracking an individual are available whenever required so they can effectively perform their duties.
*Continuous monitoring (5c):* The owner requires that the captured data is continuously monitored manually and/or by using automated tools.

***Authorized access:***
*Public access to their data (6a):* Certain countries, for example Canada and France, allow an individual to watch their own images captured by the surveillance system. Therefore, people should be able to get access to the images containing them, through a predefined procedure.
*Occasional access (6b):* As discussed in § 2.2.2, occasional access to the data might need to be given to certain public organizations; the system needs to build a mechanism to enable such access.
*Proportionate access (6c):* In order to protect the privacy of people, the owner requires that the proportionate access principle is implemented in the system and that the observers are given the minimum access to the data required to fulfill their duties. This can be achieved by implementing dynamic access control.
*Dynamic access control (6d):* The system must take the context pertinent to a situation into account when authorizing access to data so that different access levels (e.g. blurred, hiding identity of people, original images) are maintained in different situations (e.g. normal, emergency) and privacy of people is preserved to the maximum extent. In short, the access level should change appropriately depending upon the situation.
*User management (6e):* This involves all the issues related to the users of the system including user enrollment, permission assignment, changing permissions, permission revocation, user deletion etc.
*Logging (6f):* All activities performed by the observers should be securely logged, especially those permissions requested explicitly.

***Others:***
*Security of software and hardware infrastructure (7a):* It is to be ensured that the security of the underlying infrastructure is well protected against the attacks exploiting software vulnerabilities or physical access to the hardware.
*Deletion after retention period (7b):* Different countries have different regulations regarding the period of time the captured data can be kept by the operator (cf. Appendix-A). Depending upon the regulations of the region where video

surveillance takes place, the captured data must be automatically deleted as soon as the retention period expires.

Considering the concerns of each stakeholder involved in video surveillance system, it is reasonable to expect that our model has identified a comprehensive set of security and privacy requirements, though a complete set of requirements is not guaranteed. As mentioned previously, there exists a large amount of work on protecting privacy in video surveillance. The next section briefly summarizes the major types of available techniques for protecting privacy, followed by the state of the art of security research in video surveillance system and the associated challenges.

## 3.2 Privacy in Video Surveillance

A pervasive video surveillance system may be exploited by the observers for unauthorized collection of data on the activities of an individual [34]. As stated earlier, in the United Kingdom, a report discovered that observers have used video surveillance for voyeurism [96]. Possibilities for such misuse are further increased with the advent of modern video surveillance systems that facilitate rapid data retrieval enabled by indexing and searching and advanced imaging technology allowing high-resolution and zooming-in. Moreover, pervasive surveillance networks may enable linking the activities of a target in multiple video streams [87].

Considering the above-mentioned issues, several techniques to protect the privacy of the observed individuals have been proposed. In order to hide the identity of observed subjects, identity revealing sensitive areas are first determined and then removed or de-identified depending upon the approach used. Several types of techniques to hide privacy-sensitive areas have been proposed. A simple technique is to fully remove the sensitive regions; this technique not only hides the identity but in some cases also the behavior, see for example [40] [41] [139]. Another type of approach is to reduce the level of detail of privacy-sensitive areas, with the help of blurring or pixilation, leaving the subject unidentifiable yet the behavior remains recognizable, see [122] [127] [160] to name only a few. The third approach, called abstraction, is to remove the sensitive regions and replace them with dummy objects such as silhouettes or skeletons. Some of the key works in this area are [130] [61] [73]. Yet another technique proposed in literature, called scrambling, is to encrypt the sensitive regions with a key allowing the area to be decrypted only by authorized personnel possessing the key, see for instance [46] [27] [32]. As compared to other techniques, this approach offers the benefit of perfectly reconstructing the original image.

## 3.3 Security in Video Surveillance

A study of the relevant literature, reveals that many solutions, discussed below, addressing the security requirements including integrity, authentication and confidentiality have been proposed in multimedia systems e.g. video on demand and business video conferencing. However the factors involved in video surveillance systems are quite different than multimedia systems hence these solutions cannot be directly applied in video surveillance systems, although a few commonalities exist. In video surveillance systems, the security aspects are to be addressed when data is transferred from camera-to-server, server-to-server and server-to-handheld devices or monitoring room, due to the communication over public networks. We discuss here why the security requirements in video surveillance systems are important and identify the challenges to be addressed when designing security solutions for these requirements.

### 3.3.1 Integrity and Authenticity

An important security consideration is integrity protection and authentication of recorded video data. This is important for two reasons [14]: i) to accept the recordings as evidence in a court of law, and ii) to avoid framing an individual, e.g. by tempering with the recordings of a crime scene. Two major techniques to address integrity exist [128]: using cryptographic hash functions along with digital signatures or by making use of watermarks in the video recordings. Solutions proposed in multimedia systems mostly use cryptographic techniques [128] [137]. The integrity protection solution is desired to be robust against certain modifications such as scaling and compression and images should be verifiable despite such benign modifications [137]. In order to ensure authenticity, cameras need to authenticate themselves to the server. Some of the key solutions proposed in this respect require to use Trusted Platform Module (TPM) in each camera [153] [152] [151]. This approach is prohibitively expensive in terms of re-installation of existing cameras with TPM-enabled cameras. Furthermore, performance and scalability remain issues to be resolved too.

### 3.3.2 Confidentiality

Similar to integrity and authentication, there are several solutions presented for confidentiality mainly targeting multimedia applications [82] [80] [133]. In order to fulfill these requirements, the existing solutions essentially use cryptography. However, the conventional cryptographic algorithms used in these

solutions are not especially designed to encrypt video data [81]. Their usage on video data, although compressed, requires significant processing power, for instance, an MPEG-2 video stream requires a bit rate ranging between 4 to 9 Mbps [62]. Because of the huge amount of data and real-time requirement, efficient usage of cryptography is far from the desired efficiency level in conventional multimedia applications [81], whereas its usage in video surveillance introduces further challenges. In video surveillance systems, unlike multimedia applications, there are several video producers (cameras) with limited processing capabilities. A major challenge, therefore, is to devise encryption algorithms which may efficiently encrypt the large amounts of continuously produced video data, transferred in real-time to the server side, by the cameras. Another relevant issue is key management. Along with encrypting the data from each camera with a different key, the keys may also need to differ for each chunk of data, for instance different key for each 24 hours of data recorded by a camera.

A few solutions addressing confidentiality in video surveillance systems have also been proposed [126] [83] [33]. In order to protect the privacy of individuals and to ensure efficient retrieval of data, modern video surveillance systems extract metadata such as object identification, number of objects and the object types contained in the video streams in real-time [58]. This data is normally extracted at the server, therefore the server must be able to access decrypted data. Solutions proposed in [126] and [33] fail to consider this aspect and share the keys among users requiring them to collaborate when data is to be decrypted. Another reason for the server to access plain data contents is to be able to send modified video streams (low resolution, obfuscated privacy regions) to different users depending on their access authorization, discussed later in this section. Once metadata has been extracted at the server, another interesting research issue is to securely store the data along with the associated metadata in a manner that it is possible to efficiently retrieve metadata and its associated video streams later, based on query language, for example.

### 3.3.3 Authorized Access

Another important challenge which we believe requires major research effort is access authorization in video surveillance systems. Controlling the access to data is of critical importance, as the potential capabilities offered by modern video surveillance systems such as searching for an individual or an event, and monitoring the activities of an individual spanning over multiple locations [87], makes it very easy to invade the privacy of individuals. Clearly video surveillance is expected to become more pervasive and this leaves us with only two choices: either entrust the observers or to devise a mechanism for watching the watchers and minimizing the chances to use such systems abusively [28].

Similar to the above-mentioned security requirements, there exist several solutions regarding access control mechanisms for online and other payment-based video databases [23] [24] [107]. Bertino et al. [23] argue that an effective and efficient access control mechanism in video databases requires advancements in extraction of meaningful metadata. Furthermore, such mechanism must take benefit of the indexing structure used to store the video data. With advancements in indexing and metadata extraction techniques in video databases, we believe that the research efforts now need to focus on devising access control techniques for video surveillance systems. Below we discuss research efforts that discuss access control explicitly for video surveillance systems. We discuss other access control approaches relevant to video surveillance systems such as access control in multimedia databases and satellite images and compare them with our proposed model, in § 5.4.

There are only a few research attempts that focus on the challenge of access control in video surveillance. Senior et al. [130] present the idea of using multiple privacy levels in video surveillance systems where different observers are provided different levels of information and actions to be performed, depending on the access privileges of the observer. Different information levels may include, for example, access to behavioral information while hiding the identity of individuals in the video by replacing the objects contained in the video with silhouettes. Similarly different levels of actions to be performed include restrictions over playback, zooming-in and searching functions, offered by the system. The authors suggest using a privacy-preserving console manager that makes use of encryption and access control mechanisms and shows the data to the observer by revealing information components from video streams as per the authorization level of the observer. In order to use this approach, a large-scale video surveillance system requires an access control model specifically designed to meet requirements of such systems. However, the paper presents only the concept of privacy-preserving console manager without providing details of the access control mechanism.

Moncrieff et al. [87] argue that using static security policies in video surveillance is either too intrusive for privacy or it hinders the usability of the system. They identify the challenge of utilizing the video surveillance system by exposing sufficient need-specific data to the observers while preserving the privacy of people. The authors suggest that one possible way of protecting privacy in video surveillance while retaining its useful functionality is to use dynamic access control mechanisms. They propose to incorporate the context of the requester in the access authorization, where privacy is maintained using data hiding techniques in normal situations, whereas a request to data in certain situations, e.g. emergency cases, would enable the users to access full information with less focus on protecting privacy. Similar to the above-mentioned solution [130], this paper also does not provide an access control mechanism. The main contribu-

tion of this paper is to identify the challenge of a dynamic access control in video surveillance while leaving the designing of dynamic access control model as a goal to be achieved in future research. The requirements that we identified in our video surveillance model also emphasize this challenge and demand that the context of the requester is taken into consideration while granting the access. We take on this challenge of designing a dynamic access control model and propose a mechanism that preserves privacy of people without underutilizing the efficacy of the system (cf. § 3.4 and Chapter-5).

Birnstill & Pretschner [25] propose the use of usage controlled mechanism in video surveillance in order to protect the privacy of recorded people. They propose to use two different operational modes called default and alarm. The default mode aims to protect the privacy by showing only the site map view of the surveillance area with type and location of objects. The alarm mode is activated upon triggering of an alarm and shows the video streams without hiding the privacy-sensitive regions. However, the authors neither define an access control model nor do they discuss the structure and language that may be used to specify the authorization policy. They assume that the access control policy is already in place and discuss the architectural and enforcement requirements in order to apply such a policy.

To the best of our knowledge, Thuraisingham et al. [145] present the first formal access control model that targets video surveillance environments. The solution makes use of metadata extracted from video data. This metadata represents the objects and events contained in a video stream, in addition to the timestamp and location where it was captured. It presents a grammar that allows representing video streams using the associated metadata. Access privileges for observers can be specified using predefined credential expression templates based on their id, group or other characteristics, e.g. area associated with a user. The solution, however, offers a static access control model and does not allow the access privileges of an observer to be changed dynamically based on the changing situation.

Finally, in a large-scale video surveillance systems requiring occasional access by multiple public organizations such as the police and fire-brigade, management of users is also a challenge. This may require using federated identity management allowing each participating organization to manage its own users. Existing federated identity and access management solutions like SAML [98] and WS-Federation [99] may be investigated for this purpose.

Table 3.3 provides a list of future challenges in security of video surveillance systems. Each challenge refers to the related requirements given in Table 3.2. Based on our model and the discussion, it is evident that many security requirements in video surveillance systems still require further research in this domain.

Protecting the privacy of individuals without compromising the functionality of the system demands an access control mechanism that makes use of privacy enhancing technologies in order to hide the privacy sensitive regions in the video frames while making them available when required. Clearly there exists a gap demanding further research in this domain in order to satisfy the security requirements in video surveillance systems and to increase their acceptability in society.

**Table 3.3:** Future research challenges in security of video surveillance systems

| Security aspect | Future research challenges |
|---|---|
| **1. Confidentiality** | *1.1.* Novel efficient real-time encryption algorithms for large-scale video data from multiple sources (2a)<br>*1.2.* Duration-specific key management techniques for data produced by several cameras (2b)<br>*1.3.* Secure storage of video data and the associated metadata while enabling efficient retrieval (5b) |
| **2. Integrity & Authenticity** | *2.1.* Integrity protection solutions having robustness against benign modifications (3b, 3c)<br>*2.2.* Scalable and efficient authenticity mechanisms for large-scale video surveillance data (4a, 4b) |
| **3. Authorized access** | *3.1.* Multiple privacy levels in the video surveillance data, making use of existing privacy enhancing techniques, with each level accessible to different access privileges (1c, 1e, 1f, 6c)<br>*3.2.* Dynamic access control that enables preserving the privacy of people yet exposing maximum data to the observers when needed (6b, 6c, 6d)<br>*3.3.* Novel access control mechanisms utilizing the indexing structure of video data and the extracted metadata (6a)<br>*3.4.* Federated identity and access management solutions for authorizing access of video surveillance data (6b, 6c) |

## 3.4 Access Control in Video Surveillance

Out of the challenges identified above, in this thesis, we focus on addressing the challenge of dynamic access control in video surveillance systems. We chose this area due to the lack of existing research in this area and due to the requirement of an access control mechanism by our industrial collaborator, Milestone Systems, Denmark. Considering the requirements identified for authorized access in § 3.1, including proportionate access and occasional access of data, we describe a few scenarios below. In order to address the challenge of dynamic access control in video surveillance, we point out the features desired in an access control model suitable for video surveillance systems, using these scenarios.

### 3.4.1 Scenarios

Suppose a modern video surveillance system is deployed in the region of Manhattan in New York city where cameras are deployed in the major public places all over the region. The Manhattan region administration staff is allowed to regularly monitor video data recorded in different regions of the city. When granting access to video data, the information related to user and resources is also taken into consideration. To better illustrate the diverse and complex requirements of access control in video surveillance, we outline some practically relevant application scenarios below.

*Scenario-1:* Carol and Dave are working as patrolling observers in Manhattan region administration. Assume that the region of Manhattan is divided into multiple smaller regions, e.g. Manhattan_north, Manhattan_south etc. In order to reduce the privacy invasion and to minimize the possibility of voyeurism, the patrolling observers are granted access to video data of the smaller region, Manhattan_north for example, from which they are passing by, determined based on observer's current location. In normal circumstances, the observers are given lower privileged access, e.g., the faces are blurred in the video, to protect the privacy of people in the videos.

*Scenario-2:* Suppose the video surveillance system is equipped with event detection such as detection of crossing a forbidden fence, leaving luggage behind, fire detection, etc. The event detection may be implemented by exploiting techniques from the fields of pattern recognition and computer vision or by incorporating other sensors deployed on the camera locations e.g., fire detectors. Let us assume a fence is crossed at a certain location in Manhattan_south. Upon detection of this incident, an alarm goes off and system tries to find the nearest patrolling observer. Carol is currently patrolling in Manhattan_north and is the closest patrolling observer. Carol is granted access to the video data surrounding the region where the alarm is generated, in addition to the video data located in her patrolling region. Based on this incident, Carol's access privileges are elevated and she may access video data, for instance, without hiding the privacy sensitive regions of recorded people in the videos related to the alarm activated region.

*Scenario-3:* Suppose there is a fire incident reported in Times Square. Upon detection of this emergency situation, along with the regular observers assigned to this location, the nearest fire-brigade and police station, called the collaborating organizations (CO), are also informed about the event and the system allows access of data to the responding employees of the COs. Allowing access to the video data for these COs helps the emergency response teams understand the severity of the situation and to come prepared with appropriate tools and

man-power to better combat such situations.

From scenarios 1 and 2, we observe that the users are not always stationary but they may also access videos using mobile devices. Traditional video surveillance systems are typically monitored in a closed room whose access is physically controlled. Modern video surveillance systems are monitored over a variety of devices using Internet and hence require logical access control mechanisms to control access to data. This implies that the relevant information, e.g. user's location and time of access, plays an important role in granting access to videos and needs to be evaluated at the time of request evaluation. In addition to the observer's location and time of access, the videos are also associated with a particular location and time of recording. Furthermore, the information influencing the access to resources may also include occurrence of an anomalous event in a surveilled region.

The third scenario requires that the access control mechanism allows granting occasional access to video data from a specific region for responding individuals from COs based on, for instance, occurrence of an incident. Allowing access to the data to certain individuals only in case a specific incident occurs or an emergency situation, addresses the privacy concerns raised because of continuous video surveillance. Using this technique can prove to be immensely useful in public video surveillance to increase its wider acceptability in public.

### 3.4.2 Desired Access Control Features

Based on the above scenarios, we observe that changes in security characteristics may arise from resource (e.g. type of camera), user (e.g. location) and environment (e.g. incident). In order to address the challenge of utilizing the video surveillance systems by exposing sufficient need-specific data while preserving the privacy of people, access control model should effectively be able to implement the security and privacy policies. Such privacy policies may imply that only site-map view of recorded location with silhouettes, or low resolution video data with blurred faces is accessible to a user under normal circumstances, but this may be elevated to higher resolution with clearly visible faces in emergency situations. As video data contains different layers of information, different users must be given different information and set of actions to be performed on the video data, depending on the given circumstances.

From the scenarios, it is evident that the access control policies are dependent on various factors. In addition to the features related to policy specification and enforcement, certain features related to policy administration are also desired in such large-scale systems. Below we provide a list of features in terms of

functionalities both for policy specification and enforcement (i.e. 1-4), as well as for policy administration (i.e. 5 & 6) desired in an access control model for video surveillance systems.

*1- Metadata-based permissions:* Unlike the access control models which refer to objects based on their identifiers, the access control policy in video surveillance system may refer to objects not only by their identifier but also other characteristics associated with them. These characteristics include location and time when the video was recorded and contents of the data e.g., semantic objects contained within the video. Such type of information is normally captured using attributes associated with objects. Hence we need an access control mechanism which allows to use object attributes in policy specification.

*2- User context:* A user's access to resources may be dependent on several parameters associated with the user. Thus the attributes may also be associated with users such as duty-timing and location of the user. Therefore, the context of the user needs to be considered when the user is allowed access to a resource.

*3- Environmental information:* Authorization can be influenced by time of access or occurrence of an incident, e.g. fire, act of terrorism, etc. Such type of information is typically referred to as environmental information [36] and is captured via environmental attributes. The model should support incorporating such information in the access control policy.

*4- Dynamic attributes:* Attributes whose value can change quite frequently, e.g. location of user, occurrence of an incident, are called dynamic attributes. The access control mechanism must be appropriate for such dynamic attributes and must retrieve the current values of these attributes at the time of making an access control decision.

*5- Simplified Auditing:* An important administrative feature required in such systems is auditability. In order to allow auditability, the model should facilitate to review which permissions a user may exercise in what circumstances. This allows determining the risk exposure for a given employee or job position by looking at the set of permissions available to that job position or employee.

*6- Modification visualization:* Another relevant administrative feature is that the access control model should facilitate changing permissions assigned to a user and to analyze the effect of policy changes, e.g. adding or removing a privilege. It should be easy to visualize effect of a modification in the policy, i.e., who would be affected by a certain change in the policy.

In contrast to many traditional applications, video surveillance environment is typically characterized by specification of policies based on multiple user at-

tributes rather than simply user identity or role, support for content-dependent (metadata) authorizations, and authorizations being influenced by the contextual information associated with a user or environment. The requirements demand that the model must also provide features such as auditability typically offered by role-based access control solutions. Considering these features, in the next chapter, we present a general-purpose dynamic yet role-oriented access control model which is suitable for not only video surveillance systems but also other applications sharing similar requirements. In Chapter-5, we present an access control mechanism using this general-purpose model for video surveillance systems.

## 3.5 Summary

In this chapter, we presented an abstract model that allows to identify the security and privacy requirements in video surveillance systems. Our model used a two-stage process to capture requirements. In the first stage, we point out the requirements in each component of our model including data transport and monitoring from the perspective of different stakeholders, e.g. people and operating organization. The second stage remapped these requirements considering the conventional security and privacy aspects in order to remove repetitive requirements. We then discussed the existing solutions regarding these security and privacy requirements and outlined the challenges which need to be addressed. Finally, the access control features desired in video surveillance are deduced with the help of few scenarios. In the next chapter, we develop an access control model considering these features.

Chapter 4

# Attributes Enhanced Role-Based Access Control Model

In the previous chapter, we identified the features desired in access control model for video surveillance systems. Our study of existing access control models indicates that no access control model offers these features (cf. § 5.4). Motivated by these required features in video surveillance systems and the NIST initiative [74], we first develop a general-purpose access control model that combines the advantages of RBAC and ABAC in a role-centric manner (cf. § 2.4.3). The reason to develop a general-purpose access control model is to provide a model that is suitable for not only video surveillance environments but for also other applications sharing similar requirements.

This chapter presents our Attributes Enhanced Role-Based Access Control (AERBAC) model and its formal specification. We also present algorithms for two different ways in which access requests may be processed. Moreover, we evaluate AERBAC by comparing it with RBAC and ABAC, and elaborating each of these models with respect to the features required by video surveillance systems.

## 4.1 Overview of AERBAC Model

This section presents an overview of the AERBAC model. Figure 4.1 depicts our access control model and its components. The entities users, roles, objects (OBS) and operations (OPS) have the same semantics as in RBAC (cf. § 2.4.1). Users and objects in our model are associated with attributes too. We also incorporate environment attributes to fully capture the situation in which access needs to be authorized. In addition to the components of the model, the dotted-box in Fig. 4.1 represents the modules of the architectural design required to enforce this model. Below, we first describe the attributes and then discuss semantics of different components involved in AERBAC, including permissions, conditions, sessions and request evaluation.

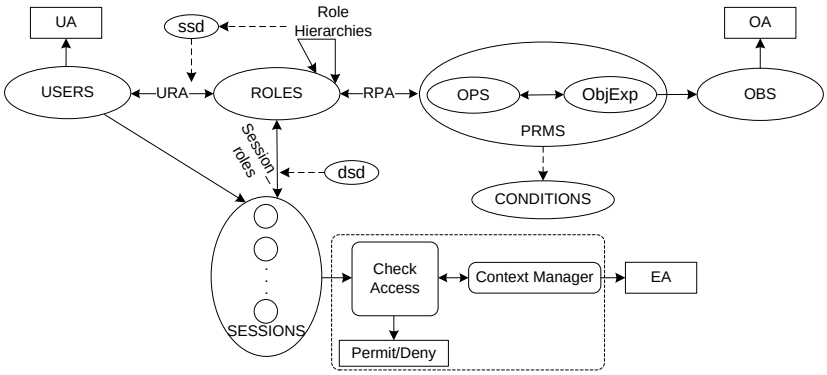


**Figure 4.1:** Attributes enhanced role-based access control (AERBAC) model

**Attributes:** Attributes capture the properties of specific entities (e.g. user). We define an attribute function for each attribute that returns the value of that attribute. Each attribute is represented by a range of finite sets of atomic values[1]. For example, the range of *branch* attribute is a set of branch names semantically relevant for the application domain. *User attributes (UA)* capture the properties of the user who initiates an access request. Examples of typical user attributes are title, affiliation, specialization, location, security clearance etc. *Object attributes (OA)* are used to define the properties of the resources protected by the access control policy. Examples of typical object attributes include type, owner, status, location, time of object creation etc. *Environment attributes (EA)* capture external factors of the situation in which the access takes place. Temperature, time of day, occurrence of an event, system mode or other information which not only pertains to a specific object or user, but may hold

[1]For handling continuous data such as temperature, the elements of the finite set comprise of ranges of intervals.

for multiple entities, at the same time, are typically modeled as environment attributes.

An attribute may be either *static* or *dynamic*. The values of *static* attributes rarely change e.g. designation, department, type etc. On the other hand, *dynamic* attributes have values that may change frequently and unpredictably, so they may even change during the lifetime of a session. This means that they may need to be checked more frequently, depending on the application requirements. Examples of such attributes include officer in command, location, occurrence of an incident etc. Dynamic attributes are also referred to as *contextual attributes* in the literature [37].

**Permissions and conditions:** Our aim is to provide a dynamic, yet easy to manage solution to enforce the access control model. In our model, this is done by incorporating attributes associated with user, objects and environment. In contrast to the traditional approaches in RBAC, the permissions in AERBAC refer to objects indirectly, using their attributes. A permission refers to a set of objects sharing common attributes, e.g. type, label or status, using a single permission, in contrast to separate permissions for each unique object (cf. § 4.3.1). This is particularly relevant in those domains where several objects share common attribute values and helps in significantly reducing the number of permissions associated with a role.

In AERBAC, a permission consists of an object expression and an authorized operation on the object set denoted by the expression. Object expressions are formed using the attributes of objects. Each permission is associated with one or more conditions, which must be evaluated to be true in order for the user to exercise that permission. A condition associated with a permission may contain attributes of all entities including users, objects and environment. In some applications, it is required to compare user and object attributes – for example, in a bank, a manager of a branch is allowed to access only those accounts belonging to his own branch. The proposed model allows to perform such comparisons using conditions.

An example of a permission, using typical attributes e.g. label, status and clearance, is: $p$= *( (oLabel(o) = 'secret'* $\wedge$ *oStatus(o) = 'active'), read)* which states that a role having this permission can perform *read* operation on the objects which are labeled *secret* and whose status is currently *active*. Here *oLabel* and *oStatus* are object attribute functions that return the values of respective attributes for a given object. Suppose that the permission $p$ is constrained by a condition $c$= *(uClearance(u) = 'secret'* $\wedge$ *time_ of_ day()* $\leq$ *uDutyExpire(u))* where *uClearance* and *uDutyExpire* are user attribute functions that return the attribute values of a given user, whereas *time_ of_ day()* is an environment attribute function. This condition implies that, in order to be granted the per-

mission *p*, the user clearance must be *secret* and the time of access must be before the end of user's *duty timing*.

**Context manager:** As mentioned above, the values of dynamic attributes can change quite frequently. The Context Manager is responsible for propagating the updated values of dynamic attributes of the users, objects and environment. Depending on the application, such a change may require re-evaluation of an already granted permission. We discuss continuous enforcement of access control for such dynamic attributes in § 5.2.1.

**Session:** A session contains a list of permissions, along with their associated conditions, assigned to the roles activated by the user. As described earlier, the permissions are different from standard RBAC permissions in terms of referring to the objects using their attributes and being tied with the conditions that are evaluated every time a permission is to be exercised. Hence, the CheckAccess function in RBAC needs to be re-defined.

**Access request:** An important consideration, in environments motivating the proposed approach, is that the user's request may also be based on the attributes of the objects. For instance, a user might want to view all objects containing some specified characteristics e.g., objects with *oType = 'classified'* and *oDept = 'admin'*. For a user request to be granted, there must exist an object expression in the user's session that denotes the requested objects, and the condition tied to that object expression must be evaluated to be true. There are different possibilities in which such a request may be evaluated and we discuss them later in the chapter (cf. § 4.2.1).

## 4.2 Formal AERBAC Model

In this section, we present the formal model that incorporates the attributes of the user, object and environment into RBAC in a role-oriented fashion. We define the sets and functions used in AERBAC in Table 4.1. The upper part of the table shows the sets and functions defined in NIST RBAC [50] (cf. § 2.4.1) which are also applicable to AERBAC. The user-to-role assignment (URA) relation captures the mapping of assignment of roles to users. Each user can create more than one sessions thereby activating a subset of roles, assigned to that user, in each session. The function session_user(se) returns the user to whom a given session *se* belongs. The function avail_session_perms(se) returns the set of permissions available to a user in a given session.

We provide further sets and functions needed for AERBAC in the lower part of Table 4.1. UATT, OATT and EATT represent sets of attribute functions for

**Table 4.1:** Sets and functions used in AERBAC

- USERS, ROLES, OBS, and OPS (users, roles, objects and operations respectively)
- URA $\subseteq$ USERS $\times$ ROLES, a many-to-many mapping of user-to-role assignment;
- SESSIONS, the set of sessions;
- user_sessions(u: USERS) $\rightarrow 2^{\text{SESSIONS}}$, the mapping of user $u$ onto a set of sessions;
- session_roles(se: SESSIONS) $\rightarrow 2^{\text{ROLES}}$, the mapping of session $se$ onto a set of roles. Formally: session_roles($\text{se}_i$) $\subseteq$ { r $\in$ ROLES | (session_user($\text{se}_i$), r) $\in$ URA};
- avail_session_perms(se: SESSIONS) $\rightarrow 2^{\text{PRMS}}$, the permissions available to a user in a session.

---

- UATT, OATT and EATT represent finite sets of user, object and environment attribute functions respectively.
- For each *att* in UATT $\cup$ OATT $\cup$ EATT, Range(*att*) represents the attribute's range, a finite set of *atomic* values.
- attType: UATT $\cup$ OATT $\cup$ EATT $\rightarrow$ {setType, atomicType}, specifies attributes as set or atomic valued.
- OBJ_EXP = Set of all object expressions formed using the language given in Table 4.2.
- COND = Set of all conditions formed using the language given in Table 4.2.
- PRMS = $2^{(\text{OPS} \times \text{OBJ\_EXP})}$, the set of permissions.
- RPA $\subseteq$ ROLES $\times$ PRMS $\times$ COND
- Each attribute function in UATT, OATT and EATT returns either atomic or set values.

$$\forall ua \in \text{UATT}.\, ua : \text{USERS} \rightarrow \begin{cases} \text{Range(ua)} & \text{if attType}(ua) = \text{atomicType} \\ 2^{\text{Range(ua)}} & \text{if attType}(ua) = \text{setType} \end{cases}$$

$$\forall oa \in \text{OATT}.\, oa : \text{OBS} \rightarrow \begin{cases} \text{Range(oa)} & \text{if attType}(oa) = \text{atomicType} \\ 2^{\text{Range(oa)}} & \text{if attType}(oa) = \text{setType} \end{cases}$$

$$\forall ea \in \text{EATT}.\, ea \rightarrow \begin{cases} \text{Range(ea)} & \text{if attType}(ea) = \text{atomicType} \\ 2^{\text{Range(ea)}} & \text{if attType}(ea) = \text{setType} \end{cases}$$

---

users, objects and environment, respectively. The notion we used for attribute representation is adapted from [66]. We use first order logic to make formal descriptions, and follow the convention that all unbound variables are universally quantified given as Range(att). Each attribute can be of type set or atomic which is determined using the attType function. Based on the type of attribute, each attribute function returns either an atomic value or a set of values, for that attribute. Attribute functions in UATT and OATT take as an argument a user and an object, respectively. Each attribute function in EATT may or may not require an argument, depending on the attribute and the target system. For

**Table 4.2:** Language to form object expressions and conditions

$\varphi ::= \varphi \wedge \varphi | \varphi \vee \varphi | (\varphi) |$ set setcompare set | atomic $\in$ set | atomic atomiccompare atomic
setcompare ::= $\subset$ | $\subseteq$ | $\nsubseteq$
atomiccompare ::= $<$ | $=$ | $\leq$ | $\neq$

To define an object expression, set and atomic are as follows:

- set::= setoa(*o*:OBS) | ConsSet
- atomic::= atomicoa(*o*:OBS) | ConsAtomic
- setoa $\in$ {oa | oa $\in$ OATT $\wedge$ attType(oa) = setType}
- atomicoa $\in$ {oa | oa $\in$ OATT $\wedge$ attType(oa) = atomicType}

For condition specification, set and atomic are as follows:

- set::= setua (session_user(*se*)) | setoa(*o*:OBS) | setea() | ConsSet
- atomic::= atomicua (session_user(*se*)) | atomicoa(*o*:OBS) | atomicea() | ConsAtomic
- setua $\in$ {ua | ua $\in$ UATT $\wedge$ attType(ua) = setType }
- atomicua $\in$ {ua | ua $\in$ UATT $\wedge$ attType(ua)= atomicType }
- setoa $\in$ {oa | oa $\in$ OATT $\wedge$ attType(oa) = setType}
- atomicoa $\in$ {oa | oa $\in$ OATT $\wedge$ attType(oa) = atomicType}
- setea $\in$ {ea | ea $\in$ EATT $\wedge$ attType(ea) = setType}
- atomicea $\in$ {ea | ea $\in$ EATT $\wedge$ attType(ea) = atomicType}

instance, in a banking system with multiple branches, an environment attribute function would require the branch name to return the value of an environment attribute, e.g., current-mode, in that branch. The role-permission assignment (RPA) relation captures permissions that are assigned to a role when a given set of conditions are fulfilled. Clearly, the permission set may change for a role if the conditions vary between requests.

As discussed earlier, permissions in AERBAC are specified using object expressions and are associated with conditions. The language to define an object expression and a condition is given in the first part of Table 4.2. The language specifies that a *set* can only be compared with another *set* while an *atomic* value can either belong to a *set* or can be compared with another *atomic* value. The second part of the table specifies how instances of *set* and *atomic* may be formed to define an object expression and a condition. In order to define an object expression, we may use only the object attributes. On the other hand, for specifying a condition, we may use attributes of all entities (i.e. users, objects, environment) each of which is either *atomicType* or *setType*. *ConsSet* and

*ConsAtomic* represent a set of constant values and a constant atomic value, respectively.

### 4.2.1 Access Decisions

The main role of the access control mechanism is to verify whether a user $u$, requesting to perform an operation $op$, on an object $o$, is authorized to do so. As mentioned earlier, a user request can either explicitly specify an object, by listing its identifier, or can implicitly denote a set of objects by using the attributes of the objects. If the user request is not for a specific object but rather a set of objects, the system must consider the given criteria to return the requested objects. Once a user submits an access request, the request is to be evaluated against the policy. The function checkAccess in RBAC needs to be modified such that it takes the user request as input, processes the request as per the format of a given request, and returns the result. In the following, we elaborate on evaluation of both identifier-based and attribute-based requests.

***a) Identifier-based request:*** In identifier-based request, the user specifies the identifier of the object to be accessed. The evaluation of this type of request is straight-forward. In this case, the input of the function checkAccess consists of a session $se$, an operation $m$, and an object $obj$. Recall that a permission consists of an object expression and an operation and is constrained by a condition. The checkAccess function returns true if and only if i) there exists a permission $p$, in the $avail\_session\_perms$ of session $se$, that contains an object expression which evaluates to true for $obj$, ii) $m$ matches $op$, and iii) the corresponding condition $c$ evaluates to true.

***b) Attribute-based request:*** Using the second form of request, the user may specify the attributes of the object in his/her request, rather than the unique identifier of an object. Specifying the object attributes in the request implies that the user wishes to access all those objects which have the specified attribute values. Attribute-based requests may be formulated in different ways. We discuss two possible ways to formulate and process such requests: i) resource query evaluation, and ii) attribute values evaluation. In resource query evaluation, the reference monitor receives an attribute-based request from the user. From this request, a query is formed to retrieve the objects from the object database. After objects retrieval, it is checked whether or not the user is allowed to access each retrieved object. Whereas in attribute-values evaluation approach, the reference monitor first evaluates whether the user is allowed to access the objects represented by the given attribute-values. If the user is authorized to do so, the objects represented by the user request are retrieved from the object database. Otherwise, the user request is denied. Figure 4.2 graphically illustrates both of

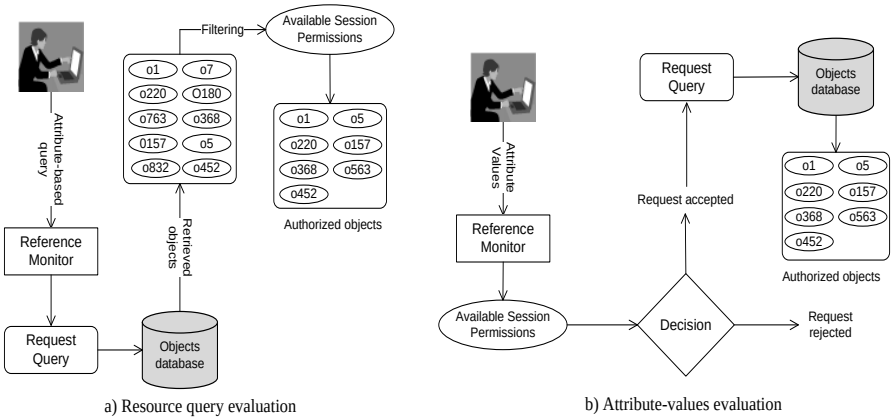


**Figure 4.2:** Attribute-based request evaluation approaches

these approaches. Below we elaborate further on these approaches and present algorithm for each approach.

***b.1) Resource query:*** In this approach, the user request contains an expression similar to the object expressions. An example user request could be: *Req = <se, (oType = 'secret' ∧ oDept = 'admin' ∧ oStatus = 'inactive'), write>* which states that the owner of the session *se* wishes to exercise the write operation on the objects denoted by the given object expression. The checkAccess function receives as input the access request *Req* and returns the authorized objects to the user, if request is granted, otherwise the request is denied. The given expression is converted to a query and the resulting objects are retrieved from the resource database. Next step is to find the applicable object expressions by matching the user's requested operation with the ones mentioned in the permission set existing in the user's session. Once the object expressions are shortlisted, they are evaluated one-by-one for each object returned by the query. If an object expression and its corresponding condition evaluate to true for an object, the object is added to the list of authorized objects that are granted to the user. Finally, the user is granted access to all those objects in the authorized list. Figure 4.3 presents the algorithm for this approach. Since the object expressions are to be evaluated for each returned object, this approach may prove to be expensive in cases where several objects are returned by the query formed based on user's request.

***b.2) Attribute values:*** An alternative strategy is to evaluate the user's request against the object expressions before retrieving the actual objects from the

**Algorithm 1**

**Input:** An access request: Req = <se, re, m >consisting of session identifier *se*, request expression *re*, and operation *m*.
**Output:** 1) Grant and return authorized objects, 2) Deny otherwise
**Begin:**

```
1: relevant_expressions = φ;
2: object_set = φ;
3: authorized_objects = φ;
4: object_set = search_objects*(re);
5: if object_set ≠ φ then
6:     for all perm<object_exp, op>∈ avail_session_perms do
7:         if m = op  then
8:             relevant_expressions ← relevant_expressions ∪ object_exp;
9:         end if
10:    end for
11:    for all object ∈ object_set do
12:        for all object_exp ∈ relevant_expressions do
13:            if evaluate†(object_exp, object) then
14:                if eval_cond‡(condition, object, session_user(se)) then
15:                    authorized_objects ← authorized_objects ∪ object;
16:                    break;
17:                end if
18:            end if
19:        end for
20:    end for
21: end if
22: if authorized_object ≠ φ then
23:    return authorized_objects;
24: end if
25: return Deny;
```

**End**

* search_objects(re) returns a set of objects existing in the resource database that are denoted by the constraints specified in expression *re*, in the request.
† evaluate(object_exp, object) returns TRUE if *object_exp* evaluates to true for the given *object*, else returns FALSE.
‡ eval_cond(condition, object, session_user(se)) returns TRUE if given *condition* evaluates to true for the given *object* attributes and the attributes of the user and the environment.

**Figure 4.3:** Algorithm for access request evaluation using resource query

resource database. In this approach, rather than providing an expression, the user specifies his/her access request by specifying the object attribute values of the desired objects. The checkAccess function receives as input the user request

*Req* and returns the objects denoted by object attribute values given in *Req*, if the request is granted, otherwise the request is denied. To process the user request, all those object expressions that use the attributes mentioned in the user's request and when the operation specified in that permission matches with requested operation, are identified in the user's session. Object expressions that include an attribute not specified by the user request are not relevant. Next, for each shortlisted object expression, the attribute functions in the object expression are given the user provided attribute values. For instance, if a user specifies the following object attribute in his/her request: *(oType = 'classified'; oDept = 'pg'; oStatus = 'active')* and suppose we find an object expression as follows: *(oType(o) = 'classified' ∧ oDept(o) ∈ {pg, ug, admin})*. Upon picking the values of the object attribute functions *oType* and *oDept* from user given attribute values we get: *('classified' = 'classified' ∧ 'pg' ∈ {pg, ug, admin})* which would evaluate to true. As soon as an object expression and its corresponding condition return true, the user's request is granted and the rest of the object expressions are ignored. When an expression returns true we form a query based on the object attribute values specified in the user request and the user is granted access to all those objects returned by the query. The algorithm for this approach is given in Fig. 4.4.

Note that we never evaluate an object expression which uses an object attribute that is not given in the user's request. This is because we replace the object attribute functions with the user given attribute values, hence any object expression involving those object attributes not given by the user cannot be evaluated. The query to get the authorized objects is formed using the object attributes mentioned in the user's request. Once an object expression returns true, this query may restrict the list of returned objects based on any additional attributes mentioned in the user's request. In the example above, the returned result is restricted based on additional object attributes *ostatus* which are mentioned in the user's request but does not exist in the expression which enables the request.

This approach is superior to resource query in terms of making an access decision by evaluating only the object expressions, without having to retrieve objects from the resource database. This is important, since many requests can be denied at this point without the overhead of object retrieval and condition evaluation. As a user is allowed to specify only the object attribute values rather than the object expression, in this form of user request, an obvious assumption is that the multiple object attributes mentioned in the user request are always combined using logical conjunction operator.

**Algorithm 2**

**Input:** An access request: Req = < se, obj_att_values, m> consisting of session identifier *se*, object attribute values *obj_att_values*, and operation *m*.
**Output:** 1) Grant and return authorized objects, 2) Deny otherwise
**Begin:**

```
1: relevant_expressions = φ;
2: authorized_objects = φ;
3: for all perm < object_exp, op > ∈ avail_session_perms do
4:     if m = op ∧ check_relevancy*(obj_exp, obj_att_values) then
5:         if evaluate† (object_exp, obj_att_values) then
6:             if eval_cond‡(condition, obj_att_values, session_user(se) then
7:                 authorized_objects = get_objects††(obj_att_values);
8:             end if
9:         end if
10:    end if
11: end for
12: if authorized_object ≠ φ then
13:    return (Grant, authorized_objects)
14: end if
15: return Deny
```

**End**

* check_relevancy(object_exp, obj_att_values) returns TRUE if the given *object_exp* uses only those object attribute functions referred in *obj_att_values*

† evaluate(object_exp, obj_att_values) returns TRUE if the given *object_exp* evaluates to true when the object attribute functions are replaced with *obj_att_values*

‡ eval_cond(condition, obj_att_values, session_user(se)) returns TRUE if the given *condition* evaluates to true for the given object attributes and the attributes of the user and environment

†† get_objects(obj_att_values) returns a set of objects existing in the resource database that satisfy *obj_att_values*

**Figure 4.4:** Algorithm for access request evaluation using attribute values

## 4.3 Evaluation

In this section, we evaluate the AERBAC model in order to illustrate the usefulness of our model. With the help of two examples, we show how our model addresses the role-explosion and permission-explosion problems, faced when using RBAC. Using these examples, we then compare our model with RBAC and ABAC with respect to the features desired in an access control model for video surveillance systems (cf. § 3.4.2).

### 4.3.1 Example Configurations

We consider two examples in our discussion. First example is about an online movie store which streams movies to its subscribed users. We consider different scenarios in this example to show how the number of roles are multiplied with the increase in the number of attributes. The second example considers a banking system, where an employee in a branch is typically allowed to access resources within his/her own branch. We configure these examples using the RBAC model to show the role- and permission-explosion issues we face while using RBAC. We then show the configuration using AERBAC to illustrate how our model strengthens RBAC by enhancing it with attributes and combines the benefits of both RBAC and ABAC.

**Example-1. Online movie store**

Scenario-1: Based on the age of the user and the rating of the movie, the system decides whether or not a user is allowed to watch a movie. An informal description of the policy rules for scenario-1 is as follows:

- Adult users can watch all movies
- Juvenile users can watch only movies with 'PG' and 'G' ratings
- Children can watch movies having 'G' ratings only

To express the abovementioned policies using RBAC, we need three roles with each having access to movies with relevant ratings. Table 4.3 lists the permissions for each role. In each role, all permissions have to be specified using identifiers of the individual movies.

**Table 4.3:** Roles and permissions in scenario-1

| **Roles** | **Permissions** |
|---|---|
| Adult | (view, movieR_1), (view, movieR_2),<br>...<br>(view, movieR_n) |
| Juvenile | (view, moviePG_1), (view, moviePG_2),<br>...<br>(view, moviePG_n) |
| Child | (view, movieG_1), (view, movieG_2),<br>...<br>(view, movieG_n) |

A user in the *Adult* role is allowed to watch movies of 'R' as well as 'PG' and 'G' ratings. Therefore, the *Adult* role is inherited from the *Juvenile* role and the *Juvenile* role is inherited from the *Child* role in order to inherit the permissions from *Juvenile* and *Child* roles, as shown in Fig. 4.5.

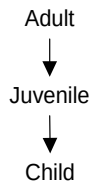


**Figure 4.5:** Role hierarchy for scenario-1

Scenario-2: Let us assume that newly released movies can only be seen by premium users. With this small change, the number of roles, needed to express this policy, are doubled. The roles we need are: *Adult-premium*, *Adult-regular*, *Juvenile-premium*, *Juvenile-regular*, *Child-premium* and *Child-regular*. Table 4.4 shows the permissions assigned to each of these roles. The role hierarchy in this case is given in Fig. 4.6.

**Table 4.4:** Roles and permissions in scenario-2

| **Roles** | **Permissions** |
|---|---|
| Adult_premium | (view, movieR_New_1), (view, movieR_New_2), ... (view, movieR_New_n) |
| Adult_regular | (view, movieR_Old_1), (view, movieR_Old_2), ... (view, movieR_Old_n) |
| Juvenile_premium | (view, moviePG_New_1), (view, moviePG_New_2), ... (view, moviePG_New_n) |
| Juvenile_regular | (view, moviePG_Old_1), (view, moviePG_Old_2), ... (view, moviePG_Old_n) |
| Child_premium | (view, movieG_New_1), (view, movieG_New_2), ... (view, movieG_New_n) |
| Child_regular | (view, movieG_Old_1), (view, movieG_Old_2), ... (view, movieG_Old_n) |

Scenario-3: Let us assume that out of the newly released movies, few chosen movies can be watched by regular users, but only during promotional season. To express this environmental condition, we need three additional roles (*Adult_regular_promo, Juvenile_regular_promo, Child_regular_promo*) which will be allowed to be activated by users only during promotional season, using role-activation constraints in RBAC. Table 4.5 shows permissions for these roles. Figure 4.7 provides the modified role hierarchy for this scenario.

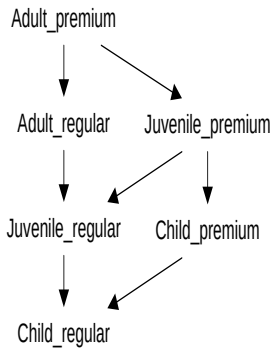


**Figure 4.6:** Role hierarchy for scenario-2

**Table 4.5:** Roles and permissions for promotional period in scenario-3

| **Roles** | **Permissions** |
|---|---|
| Adult_regular_promo | (view, movieR_Chosen_1), (view, movieR_Chosen_2), ... (view, movieR_Chosen_n) |
| Juvenile_regular_promo | (view, moviePG_Chosen_1), (view, moviePG_Chosen_2), ... (view, moviePG_Chosen_n) |
| Child_regular_promo | (view, movieG_Chosen_1), (view, movieG_Chosen_2), ... (view, movieG_Chosen_n) |

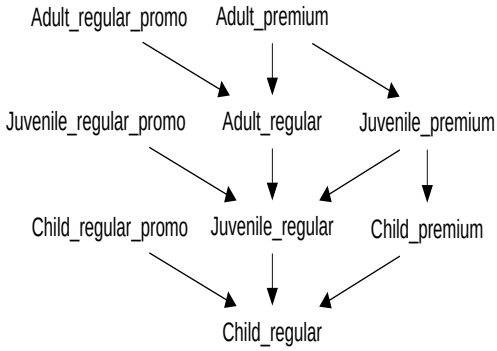


**Figure 4.7:** Role hierarchy for scenario-3

#### 4.3.1.1 Role- and Permission-explosion in RBAC

In RBAC, the number of roles increases exponentially as the number of user and environment attributes grow [54], as we can observe in the above discussion. Note that the object attributes do not influence the number of roles since the object attributes are transformed into user attributes to distinguish between users who may access an object with a specific attribute value. For instance, in the above example, movies with rating 'R' are accessible to users whose age is greater than 18. In worst case, the number of roles required for $K$ attributes with $V$ values each can be calculated as $K^V$. However, the number of attribute values is not necessarily the same for each attribute. For example, in the above-mentioned scenarios, there are two user attributes (*age*, *member-type*) and one environment attribute (*promo-period*). The possible values for *age* attribute in this example are: >10, >13 and >18. The values of *member-type* could be either 'regular' or 'premium', whereas the values of *promo-period* (environment attribute) could be either 'yes' or 'no'. Therefore, the number of maximum roles in worst case, for given example, can be calculated as:

$\Rightarrow V(age) \times V(member\text{-}type) \times V(promo\text{-}season)$
$\Rightarrow 3 \times 2 \times 2 = 12$

Note that *V(att)* above represents the possible number of values for an attribute of a user or environment. In scenario-3, since the *promo-period* affected only regular users, thus the total number of roles was 9, when considering *age*, *member-type* and *promo_period* attributes, as shown in Fig. 4.7.

Assuming that roles are formed using all user and environment attributes, a general formula to calculate the maximum number of roles in worst case is:

$$roles = [V_{UA_1} \times V_{UA_2} \times \ldots \times V_{UA_n}] \times [V_{EA_1} \times V_{EA_2} \times \ldots \times V_{EA_m}]$$

$$roles \;=\; \prod_{i=1}^{n} V_{UA_i} \times \prod_{j=1}^{m} V_{EA_j} \tag{4.1}$$

where $V_{UA_i}$ and $V_{EA_i}$ represent the number of possible values for a user and environment attribute, respectively.

On the other hand, the number of permissions in a role is directly proportional to the number of objects relevant to that role. The following expression represents the relation between the permissions and objects in a role:

$$\uparrow permissions_{role_i} \;\propto\; \uparrow objects_{role_i} \tag{4.2}$$

This expression implies that the increase in the number of permissions in a role is directly proportional to the increase in the number of objects relevant to that role. Thus, with the growth in the number of objects, the number of permissions also grows since the permissions in RBAC are specified using identifiers of the objects. In the above example, the number of permissions in each role depends on the number of movies related to that role.

### Configuration using AERBAC

To address the issues of role- and permission-explosion in RBAC, AERBAC integrates roles and attributes in a novel way by using the attributes of the objects in the permissions, rather than using identifiers of individual objects, and attributes of all entities in conditions. Below, we configure the scenarios discussed above for the online movie streaming example using AERBAC model.

Scenario-1: Using AERBAC, the policies in scenario-1 can be specified as given in Table 4.6. Rather than using the identifier of the objects, we use object attribute function *Rating(o:OBS)* that takes a movie object as input and returns the rating of that movie. This allows us to refer to all movies having a specific rating using a single permission. There are no conditions associated with permissions as no user and environment attributes are used in this scenario.

**Table 4.6:** Roles and permissions using AERBAC in scenario-1

| **Roles** | **Permissions** | **Conditions** |
|---|---|---|
| Adult | (view, Rating(movie) = 'R') | None |
| Juvenile | (view, Rating(movie) = 'PG') | None |
| Child | (view, Rating(movie) = 'G') | None |

Scenario-2: Using AERBAC, the policies in scenario-2 can be specified as given in Table 4.7. The object attribute function *Release(o:OBS)* returns whether a given movie is new or old. The user attribute function *Member_type(u:USERS)* returns whether a given user is a regular or premium user. The permissions allowing access to newly released movies are constrained by conditions which ensure that only premium users can watch these movies.

Scenario-3: Using AERBAC, the policies in scenario-3 can be specified as given in Table 4.8. The conditions in this scenario ensure that the chosen movies can be watched by all users only if it is promotional season. Here, the object attribute function *Chosen(o:OBS)* returns whether a given movie has been chosen to be shown during promotional season. Whereas *Promo_season()* is an environment attribute function that returns whether the promotional season is active or not.

Using AERBAC, the hierarchy of roles stays the same for all three scenarios as

**Table 4.7:** Roles and permissions using AERBAC in scenario-2

| Roles | Permissions | Conditions |
|---|---|---|
| Adult | (view, Rating(movie) = 'R' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'R' ∧ Release(movie) = 'new') | (Member_type(user) = 'premium' |
| Juvenile | (view, Rating(movie) = 'PG' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'PG' ∧ Release(movie) = 'new') | (Member_type(user) = 'premium' |
| Child | (view, Rating(movie) = 'G' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'G' ∧ Release(movie) = 'new') | (Member_type(user) = 'premium' |

given in Fig. 4.5.

Note that, in AERBAC, we use an attribute function for each attribute to get the value of that attribute. For instance, *member-type* is a user attribute whereas *Member_ type(u:USERS)* is a user attribute function that returns the value of this attribute for a given user. As compared to RBAC approach, we do not need to make frequent changes in the permission-set assigned to a role in AERBAC each time a new movie needs to be added or removed. For example, to remove a chosen movie from the list of objects accessible to a role, we would simply change the value of *chosen* attribute for that object rather than making changes in the permission-set of the role. Thus our model helps to avoid making frequent changes in the permission-set of a role, in cases where the objects are added/removed frequently.

**Addressing role- and permission-explosion**

As shown in the example configuration above, we address the problems of permission-explosion and role-explosion by using object expressions and conditions. Using AERBAC, the number of roles does not increase with the growth in the number of user or environment attributes. As shown in the example, the roles in AERBAC depend on the job-function/task and, unlike RBAC, we do not need to create similar roles based on user or environment attributes. The following expression represents the relation between roles and attributes of the user and the environment:

$$\uparrow roles \quad \not\propto \quad \{\uparrow U_{AV} \quad \vee \quad \uparrow E_{AV}\} \tag{4.3}$$

**Table 4.8:** Roles and permissions using AERBAC in scenario-3

| Roles | Permissions | Conditions |
|---|---|---|
| Adult | (view, Rating(movie) = 'R' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'R' ∧ Release(movie) = 'new') | Member_type(user) = 'premium' |
| | (view, Rating(movie) = 'R' ∧ Chosen(movie) = 'yes') | Promo_season() = 'yes' |
| Juvenile | (view, Rating(movie) = 'PG' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'PG' ∧ Release(movie) = 'new') | Member_type(user) = 'premium' |
| | (view, Rating(movie) = 'PG' ∧ Chosen(movie) = 'yes') | Promo_season() = 'yes' |
| Child | (view, Rating(movie) = 'G' ∧ Release(movie) = 'old') | None |
| | (view, Rating(movie) = 'G' ∧ Release(movie) = 'new') | Member_type(user) = 'premium' |
| | (view, Rating(movie) = 'G' ∧ Chosen(movie) = 'yes') | Promo_season() = 'yes' |

$U_{AV}$ and $E_{AV}$ above represent number of attribute values for user attributes and environment attributes, respectively. The expression implies that the increase in the number of roles is not directly proportional to the increase in user or environment attribute values.

In AERBAC, the number of maximum permissions does not depend on the number of objects rather it depends on the number of object expressions which are formed using object attributes. The number of maximum object expressions which can be used in a role's permissions, for the given example, can be calculated as:

$$\begin{aligned} \Rightarrow obj\_exps_{role_i} &= V(rating) \times V(release) \times V(chosen) \\ \Rightarrow obj\_exps_{role_i} &= 3 \times 2 \times 2 = 12 \end{aligned}$$

Note that *V(att)* above represents the possible number of values for an object attribute. Making an assumption, similar to the one in equation (4.1), that object expressions in a role are formed using all object attributes, a general formula to calculate the maximum number of object expressions in a role is:

$$obj\_exps_{role_i} = \prod_{i=1}^{n} V_{OA_i} \tag{4.4}$$

where $V_{OA_i}$ represents the number of possible values for an object attribute. Using the above equation, the maximum number of permissions in a role can be calculated as follows:

$$permissions_{role_i} = obj\_exps_{role_i} \times operations \tag{4.5}$$

where *operations* specify the number of elements in the set OPS.

In the above example, we achieve significant reduction using AERBAC, as compared to RBAC, in the number of roles and the number of permissions. Notice that the number of roles in worst case, in RBAC, should not be confused with the number of object expressions in worst case, in AERBAC. In RBAC, we need to have 9 roles – each having a large number of permissions depending on the number of objects relevant to that role. Whereas, in AERBAC, we need three roles and the total number of permissions are 9 in all three roles.

**Example-2. Banking system**

AERBAC allows to directly compare object attributes with user attributes using conditions. Comparing object attributes directly with user attributes may significantly reduce the number of permissions specified for a role as well as the number of roles. Consider the example of a banking system. A bank typically has several branches hence the possible values for *branch* attribute includes hundreds of branch locations in that bank. An employee, say *Manager*, of a branch is allowed to access account information of customers related to his own branch only.

RBAC deals with such a situation by creating a separate role for each position of each branch, e.g., *Manager-branch-A*, *Manager-branch-B* etc. Sample roles and permissions for this example are given in Table 4.9. The permissions in each role specify the allowed operations on the specific objects existing in that branch.

**Configuration using AERBAC**

Using AERBAC, we may deal with this issue by directly comparing object and user attributes in the condition associated to a permission. Table 4.10 gives set of permissions for the *Manager* role using AERBAC. In this example, we use three object attribute functions: *oType(o:OBS)*, *oStatus(o:OBS)* and *Object_ branch(o:OBS)* which return the type, status and branch name for a given object, respectively. The user attribute function, *User_ branch(u:USERS)* returns the branch name for a given user who has placed an access request. Using AERBAC, we do not need to create separate *Manager* role for each branch,

**Table 4.9:** Roles and permissions in example-2

| Roles | Permissions |
|---|---|
| Manager_branch_1 | (read, branch_1_type_1_object_1),<br>(read, branch_1_type_1_object_2),<br>...<br>(read, branch_1_type_1_object_n) |
| | (write, branch_1_type_1_active_object_-1), (write, branch_1_type_1_active_object_2), ...<br>(write, branch_1_type_1_active_object_n) |
| Manager_branch_2 | (read, branch_2_type_1_object_1),<br>(read, branch_2_type_1_object_2),<br>...<br>(read, branch_2_type_1_object_n) |
| | (write, branch_2_type_1_active_object_-1), (write, branch_2_type_1_active_object_2), ...<br>(write, branch_2_type_1_active_object_n) |

rather we use condition to check whether the branch of the user accessing an object is same as the branch to which that object is associated.

**Table 4.10:** Roles and permissions using AERBAC in example-2

| Roles | Permissions | Conditions |
|---|---|---|
| Manager | (read, oType(object) = '1') | (User_branch(user)<br>= Object_branch(object) ) |
| | (write, (oType(object) = '1' ∧<br>oStatus(object) = 'active')) | (User_branch(user)<br>= Object_branch(object) ) |

We can see that the administrative complexity using AERBAC is much reduced for the examples given above. These are quite typical examples in many real-world applications. In practice, security architects solve these problems by combining RBAC with other solutions such as context-based or ABAC. The resulting solution is an ad hoc model which requires specialized administration for each such application [56, 125]. Moreover, solutions which extend RBAC to deal with the issues discussed above have also been proposed. However, our model is the first one to include object expressions in the permissions rather than identifiers or types/groups of objects. We discuss these solutions and compare them with AERBAC in § 4.4.

### 4.3.2 AERBAC Features

In this section, we evaluate AERBAC against the desired features, identified in last chapter (cf. § 3.4.2), for an access control model in video surveillance system. In the light of above example configurations, we compare the AERBAC model to RBAC and ABAC models and discuss which of these features are offered by each of these models.

**1- User context:** One of the factors contributing to role-explosion in RBAC is the inclusion of user attributes into the access control decisions [54]. Users in the same job role may be permitted to perform the same operations but on different sets of objects which are decided based on user's attributes or their relationship with the objects. For instance, as seen in the banking system example earlier, a user in the *manager* role should be allowed to access accounts related to his own branch only. Another example is related to health-care system where a doctor is allowed to access data relevant to his own patients, rather than all patients. This may result in creating a separate role, for each possible value of such attribute, with the same operations but different objects, causing the role-explosion issue.

ABAC offers flexibility and can easily incorporate attributes associated with the users [38]. AERBAC also allows incorporating user attributes into the access control model. Contrary to RBAC, AERBAC does not cause the problem of role-explosion because of associating conditions with the permissions, as shown in the configuration of the online movie streaming, and banking system examples above.

**2- Environmental information:** As seen in the example configurations, RBAC cannot easily handle environmental constraints such as time and the occurrence of an event and hence does not provide a fine-degree of granularity as demanded by many applications [74]. Standard RBAC [50] allows enforcing constraints either on the assignment of roles to users or on activation of roles in a session. In certain cases, it may be required to put constraints on the individual permissions assigned to a role. For instance, a user may be allowed to activate *manager* role in a bank, however the permission to update accounts in *manager* role may only be exercised during 1400 – 1600 hrs. In order to enforce such constraints and provide finer-grained access, standard RBAC requires creating separate roles which may be activated during specified time periods. Creating separate roles for such contextual constraints has two drawbacks. First, it causes the role-explosion issue due to the creation of separate roles with few permissions and users assigned. Second, it requires reactivation of such roles each time a user wants to exercise the permission in that role when the contextual constraints are fulfilled.

Dynamically changing environmental constraints can be easily handled using ABAC approach [38]. It allows representing such constraints using a rule-based policy which enables fine-grained access to resources. In AERBAC, we add environment attributes to capture such information. By using conditions to enforce such constraints at permission-level, AERBAC provides a mechanism to incorporate these dynamically changing attributes in a role-centric manner yet without requiring to create a large number of roles.

**3- Metadata-based permissions:** In RBAC, the permissions comprise of operations that can be performed on objects. The objects are specified using their identifiers which causes permission-explosion in applications having a large number of objects, as seen in the online movie streaming example (cf. § 4.3.1).

In AERBAC, the permissions comprise of operations and object expressions which are formed using object attributes. Using object expressions reduces the number of permissions to be specified in a role, as illustrated in the online movie streaming example. Referring to objects based on their attributes also allows permissions to denote those objects which will be created in the future.

**4- Dynamic attributes:** In RBAC, the value of an attributes is verified at the time of creating a session. For instance, the value of *promotional-period* in Scenario-3 of Example-1 above, is checked at the time of activating *adult_ - regular_promo* role. Once a session is created, the user can exercise all the permissions in that session without considering the fact that the value of a dynamic attribute can be changed during the lifetime of a session. Contrary to RBAC, ABAC retrieves the current values of dynamic attributes at the time of making an access control decision while evaluating a relevant policy rule. By associating conditions with permissions, AERBAC also verifies the values of dynamic attributes, mentioned in that condition, every time a user requests to exercise a particular permission.

**5- Simplified auditing:** RBAC permits simplified auditing by providing a convenient mechanism to review the permissions available to a user by having a role assigned to a user [54].

When ABAC is used in a large organization having a large number of policy rules, it may not be practically feasible to audit what permissions have been granted to a user. In ABAC, any combination of attributes may essentially grant an access and hence it requires analyzing all policy rules with an exhaustive enumeration of attributes used in each policy rule [38]. AERBAC makes it simpler to audit what permissions may be granted to a user because of being role-centric yet constraining permissions with conditions. When auditing for a particular position or employee, we need to consider only the policy rules given in the roles assigned to that position or employee.

**6- Modification visualization:** RBAC enables security administrators to visualize the users who will be affected by adding, deleting or modifying a permission [48] due to the permission being assigned to a role and the role to users.

One of the issues in the ABAC approach is that the consequences of a newly added or removed policy rule are not easy to visualize [48]. It is not clear what set of users will be effected by a change in the policy. For instance, removing a policy rule may essentially affect those users whom we wish to remain authorized to access a particular resource, but they are no more authorized since a policy rule is removed. In AERBAC, it is relatively easy to visualize what is the impact of adding or removing a policy since policy specification is at the level of roles. Therefore, a change in policy can effect only those users who are assigned to a role being modified.

#### 4.3.2.1 Additional Features

Besides the features desired in the access control mechanism for video surveillance systems, AERBAC model also offers following features by virtue of being a role-centric approach.

**7- Privilege management:** Privilege management deals with provisioning and revocation of privileges to/from a user. Controlling all access through roles decreases the overall cost of security management and makes administration of permissions efficient in terms of time and effort [39]. Roles represent job functions and can be well understood by their names. Once engineered, the provisioning or revocation of roles assigned to a user becomes an easy task and can be assigned to users even by non-expert personnel. This is where the RBAC approach really starts providing economic benefits.

In contrast, assigning and revoking of privileges in ABAC is considered a complex task [48], as compared to RBAC, since revoking a user attribute can have implications on other privileges that the administrator wishes to remain assigned to the user. Although AERBAC involves the overhead of assigning user attributes too when assigning a role to a user. However, the assignment of attributes and their values is dependent on the specific role being assigned and hence becomes a simple task. Revoking a role in AERBAC de-assigns all the permissions associated with that role, even though the user still holds rest of the attributes needed to exercise those permissions.

**8- Separation of duty (SoD):** In order to avoid conflict of interest, SoD is one of the most common constraint used in large-scale organizations. RBAC facilitates enforcement of SoD constraint [39] by defining conflicting roles which

should never be assigned together to a user (static SoD) or activated together by a user (dynamic SoD). In a similar manner, SoD constraints can be enforced in AERBAC too. On the other hand, enforcing SoD constraint in ABAC requires to use negative privileges which does not only causes difficulty in terms of policy management but it may also lead to policy conflicts demanding for a mechanism to resolve such conflicts.

**9- Enforcement of least privileges:** The principle of least privilege or need-to-know helps ensure that users have sufficient permissions needed to perform their duties while not being overentitled. This principle guarantees required productivity while reducing the security risk caused by an individual. The RBAC and AERBAC models provide an effective mechanism to enforce the principle of least privilege by assigning users to roles (the need element) and roles to permissions (the know element). By enforcing this principle, the risk of unauthorized system access is greatly minimized. No methodological mechanism to enforce the principle of least privilege exists in ABAC.

**Table 4.11:** Comparing AERBAC with RBAC and ABAC

| **Features** | **RBAC** | **ABAC** | **AERBAC** |
|---|---|---|---|
| 1- User context | ✗ | ✓ | ✓ |
| 2- Environmental information | ✗ | ✓ | ✓ |
| 3- Metadata-based permissions | ✗ | ✓ | ✓ |
| 4- Dynamic attributes | ✗ | ✓ | ✓ |
| 5- Simplified auditing | ✓ | ✗ | ✓ |
| 6- Modification visualization | ✓ | ✗ | ✓ |
| 7- Privilege management | ✓ | ✗ | ✓ |
| 8- Separation of duty | ✓ | ✗ | ✓ |
| 9- Least privileges | ✓ | ✗ | ✓ |

Table 4.11 gives a summary of the features offered by RBAC, ABAC and AERBAC models, as discussed above. A tick-mark implies that the model supports the given feature, whereas a cross-mark in the table means that the given feature is not well-supported by the model. For example, though RBAC can be used to handle environmental information, it requires creating a large number of roles causing administrative issues.

### 4.3.3 Limitations of AERBAC

In AERBAC, we attempt to combine the benefits of both RBAC and ABAC while avoiding their respective disadvantages. Although AERBAC achieves many features desired in an access control model, AERBAC has a few limitations too. Some of these limitations are introduced by AERBAC, e.g. complexity of attribute management, while inheriting few from RBAC such as ambiguity in role hierarchies. Below we discuss these limitations.

**1- Role and attribute engineering:** The process of developing a role structure for a target organization is referred to as role engineering. This seemingly simple task of role engineering is one of the most challenging and time consuming part of implementing RBAC in an organization. The role engineering process also involves identifying the exact set of permissions suitable for the organization. As AERBAC is an extension of RBAC and incorporates attributes into RBAC, this process also involves figuring out the attributes in AERBAC, in addition to roles, adding further complexity in the role engineering process.

It is to be noted, however, that the overhead of attribute engineering offers several advantages too. As we discussed in the online movie streaming example (cf. § 4.3.1), AERBAC incorporates user and environment attributes in a manner that reduces the role-explosion. Similarly, object attributes are used to address the permission-explosion problem. This implies that once the roles, attributes and permissions are figured out, the implementation of the solution using AERBAC is not hard due to significantly less number of roles and permissions.

**2- Complex permission management:** After the completion of role engineering and implementation, the management phase deals with day-to-day operations such as maintaining the role-permission assignments and assignment of roles and other attributes to the users. When assigning roles to a user, AERBAC also involves assignment of relevant attributes to the user along with the role, rather than simply the role as opposed to RBAC. However, when comparing to RBAC we observe that offering the same set of permissions in RBAC typically requires more number of roles and complex role hierarchies, as shown in scenario-2 & 3 in the online movie streaming example presented earlier (cf. § 4.3.1).

Similarly, when new objects are added in the object database using AERBAC, we need to associate the newly added object with attributes too. However, comparing to RBAC, we do not need to explicitly add a permission in those roles which may access this object, because of using attribute-based permissions in AERBAC. Due to this, the role-permission assignments are rarely modified.

**3- Permission overview:** As the permissions in AERBAC consist of object expressions and operations, therefore, when reviewing the permissions accessible to a user, we get the object expressions (containing attributes) denoting the objects rather than the object identifiers, as compared to RBAC. In order to get the identifiers of the objects available to a user, we may retrieve those objects from the object database by forming queries using the given object expressions in the permissions. Each object expression will form the maximum number of objects available to a user but constrained by the condition associated with each permission.

Another possibility is to pre-compute the list of objects accessible to a role, based on permissions in that role. However, this may prove to be expensive, as with addition of each object, the permissions in a role have to be re-evaluated to check if the added object is accessible to a role. Note that, the condition associated with the permission, granting the access to the objects, will still need to be evaluated at run-time in order to verify the dynamic user attributes, e.g. location, and environment attributes, e.g. system-load.

**4- Role and attribute semantics:** There must exist consensus among the stakeholders involved in a large enterprise or cross-organization implementation of AERBAC. Each stakeholder must associate the same set of permissions for a given role and semantics for a particular attribute. Reaching such an agreement may be a challenge especially in applications having a large number of roles and attributes.

**5- Ambiguity in role hierarchies:** Creating role hierarchies in AERBAC requires to have a clear understanding of the complex job function hierarchies in a given organization. Using role hierarchies may result in under- or over-entitlement of a user and hence requires studying the consequences of the developed role hierarchy. For instance, it might seem reasonable to inherit the role *project-manager* from the role *programmer* thereby inheriting the permissions assigned to the *programmer* role. However, a project manager might not have the required technical knowledge thus allowing him to update executables violates need-to-know principle (over-entitlement).

## 4.4 Comparison with Related Work

In response to the NIST initiative, Jin et al [67] present the first formal access control model called Role-centric Attribute-Based Access Control (RABAC) model using the role-centric approach. They extend RBAC with user and object attributes and add a component called permission filtering policy (PFP).

The PFP requires specification of filtering functions in the form of Boolean expression consisting of user and object attributes. Their solution is useful to address the role-explosion problem and as a result facilitates user role assignment. However, their approach does not incorporate environment attributes and is not suitable for systems involving dynamic attributes, e.g., location and time, due to verification of such attributes at the time of session creation only. Also, our approach is significantly different in the sense that we make a fundamental modification in RBAC by using attributes of the objects in the permissions, addressing the issue of permission-explosion, faced while using RABAC. Alshehri et al. [8] present an access control model that integrates attributes and roles. It is different than AERBAC in that it uses roles as simply user attributes and roles are not associated with permissions. Hence this approach does not take full benefit of the notion of role and does not allow inheritance of permissions – due to absence of role hierarchies – and enforcement of separation of duty constraints. Moreover, there is no concept of session rather each object is assigned a policy which is evaluated when access to that particular object is requested, similar to the mechanism used in access control lists. Huang et al [63] present a framework to integrate RBAC with attributes. The approach consists of two levels: underground and aboveground. The underground level makes use of attribute-based policies to automate the processes of user-role and role-permission assignment. The aboveground level is the RBAC model, with addition of environment attributes, constructed using attribute-based policies. Their work is different than AERBAC in that it focuses on automated construction of RBAC. Xu and Stoller [158] focus on migration of RBAC-based systems to ABAC in order to avoid limitations of RBAC. They present a solution to mine attribute-based policies from an already configured RBAC model.

As discussed in § 2.4, several efforts have been reported which extend RBAC to include the context of access [88, 92, 60, 162, 71, 68, 22, 119, 75, 29]. However these approaches do not allow comparing user and object attributes as constraints on permissions. Most of these solutions typically require creation of a large number of closely related roles, causing the role-explosion problem. Ge et al. [55], and Giuri et al. [57] focus on resolving the issue of role-explosion by providing the mechanism of parametrized privileges and parametrized roles. However, the permissions in these solutions refer to objects using their identifiers. Few approaches propose a variant of RBAC categorizing the objects into groups or types in an attempt to resolve the permission-explosion issue [88], [35], [69]. Grouping the objects allows to associate a single attribute with each object. The permissions are then specified using the group attribute – referred to as object roles in [36] & [88], views in [69] and object classes in [35] – where each permission refers to a set of objects in that group. Moreover, as the number of object attributes grow, the number of groups increase exponentially. This makes the task of policy administration cumbersome as a security administrator needs to form the potential groups in advance, based on all possible values

of the object attributes.

Another area of research relevant to AERBAC is content-based access control, where access to a resource is dependent on the information contained within the resource. Prior literature mainly uses attribute-based approaches to handle this requirement [24], [4]. However, these approaches suffer from the ABAC limitations, discussed earlier. Using a combination of roles and attributes may help in simplifying the management and policy modification, as discussed in § 4.3.

## 4.5 Summary

This chapter presented a general-purpose access control model that integrated the RBAC and ABAC models in order to bring together the features offered by both these models. Contrary to the traditional RBAC approaches, the permissions in AERBAC consist of operations and object expressions which are formed using object attributes. The object expressions allowed us to represent a set of objects using each permission and enabled content-based access control. The model is context-aware since the condition associated to every permission is verified each time a permission is requested. Two distinct algorithms to evaluate access requests were also presented. It was demonstrated that the problems of role-explosion and permission-explosion faced in RBAC are resolved in AERBAC. An evaluation of AERBAC model against RBAC and ABAC is also provided with respect to the access control features desired in video surveillance systems. The chapter also discussed the limitations of AERBAC model. Finally a comparison of AERBAC was performed with the other solutions proposed as response to NIST initiative and certain other relevant approaches.

Chapter 5

# Access Control in Video Surveillance

As discussed in the earlier chapters, modern video surveillance systems equipped with advanced functionalities, e.g. semantic object-detection and rapid data retrieval, allow the observers to traverse the data in an efficient way. Such systems give substantial powers to the observers who may profile the activities of an individual which results in compromising the privacy of people recorded by the system. We have also discussed the camera and video characteristics (cf. § 2.2.2) and observed that the semantic objects and events are extracted from the videos; such information as well as the location and time of recordings, called metadata, is stored along with the videos, using MPEG-7, for example.

In contrast to the traditional systems whose access is physically controlled, as they are watched in a closed monitoring room, the videos in modern systems can also be accessed ubiquitously over small hand-held devices. Due to the potential capabilities offered by modern video surveillance systems such as searching and tracking the activities of an individual spanning over multiple locations [87], it becomes critically important to control the access to data in such systems. To develop an access control mechanism, earlier in § 3.4.2, we deduced the features desired for access control in video surveillance. Table 5.1 presents a summary of these required features. Our study of existing access control models, proposed for video surveillance (cf. § 3.3.3) and related domains such as multimedia applications (cf. § 5.4), indicates that these models do not meet our requirements.

**Table 5.1:** Summary of features desired for access control in video surveillance

| Feature | Description |
|---|---|
| Metadata-based permissions | The access control mechanism should allow specification of permissions based on metadata information, e.g. camera deployment area, semantic objects contained in video, etc. |
| User context | The context of the user such as user's location and response-area needs to be considered when determining user's access request. |
| Environmental information | Environmental information such as occurrence of an incident or time of access may also influence an access control decision. |
| Dynamic attributes | As dynamic attribute values, e.g. user's location, occurrence of an incident, can change quite frequently; the access control mechanism must consider the current values of these attributes when making an access control decision. |
| Simplified auditing | The mechanism should facilitate to review which permissions a user or role may exercise in what circumstances. |
| Modification visualization | Visualizing the effect of a modification in the policy, i.e. which users will be affected by a change, must be facilitated by the mechanism. |

In this chapter, we present a Role-Oriented Access control Mechanism for Video Surveillance systems (ROAMVS) which is based on the AERBAC model discussed in the previous chapter. This instantiation necessitates a discussion of the following artifacts of the access control mechanism. First, we extend AERBAC with spatial and temporal constraints and define how subjects and objects in video surveillance systems may be specified using the proposed mechanism. Secondly, the actions such as read and write, used in conventional systems, are not relevant for video surveillance data. In order to provide actions semantically relevant for video surveillance data and to enable multilevel access control that reveals different levels of information to different users, we define privilege modes by combining video properties with actions. Finally, ROAMVS allows derivation of permissions from explicitly stated ones, due to hierarchical relations between the attributes of different entities, in addition to role hierarchies. We have also developed a prototype implementation of the proposed mechanism using eXtensible Access Control Markup Language (XACML) [102] to demonstrate the feasibility of our approach in video surveillance applications.

## 5.1 Access Control Mechanism

As stated earlier, time and location play an important role in determining an access control decision. Our proposed mechanism allows to use spatial (location) and temporal constraints in the access control policy. In this section, we first describe a summary of how location and temporal constraints can be formed. These constraints are then used in defining resource object expressions

(cf. § 5.1.2) and in specification of conditions (cf. § 5.1.5) associated with permissions of a role. We then describe the characteristics of objects to be protected and users to be authorized, and formalize these concepts. We also discuss formation of privilege modes that consist of video properties and actions and show creation of roles and permissions following our ROAMVS approach.

## 5.1.1 Representing Location and Time

Due to the relevance of location and time in access control, the incorporation of such information into access control has been a subject of significant bodies of work [5, 12, 21, 77, 119]. These models introduce the notions to represent location and time and the operators that can operate on location or time, which are fairly similar across models. This section briefly summarizes the commonly used concepts necessary for formal representation of time and location and defines the formation of spatial and temporal constraints. The notation we use to formalize location and time uses set theory conventions and is adapted from access control models presented in [5, 12, 21].

### 5.1.1.1 Spatial Constraints

Location of a user or object can be obtained through a trusted device. For instance, GPS can be used to accurately collect the coordinates of a mobile user. Collection of location information is beyond the scope of this work, rather we discuss the use of such information once collected. The location information is of two types: physical and logical. *Physical location* is the raw geographic location returned by the device, e.g. represented as a point in a three-dimensional geometric space. *Logical location* is application-dependent and is a symbolic representation against a group of physical locations. Examples of logical locations are New York, Times Square, engineering lab etc.

Let $\mathcal{PL}$ be the set of all physical locations and $\mathcal{LL}$ be the set of all logical locations. In order to convert physical locations to logical locations and vice-versa, mapping functions are defined. Let $\Omega$ and $\Psi$ be the mapping functions that define the correspondence from elements of $\mathcal{PL}$ to the elements of $\mathcal{LL}$ and from elements of $\mathcal{LL}$ to the elements of $\mathcal{PL}$, respectively. For instance, applying the function $\Omega$ to the physical position of a camera might return Times Square from the $\mathcal{LL}$ set. Let $\mathcal{FL}$ be the set of location functions where each function receive an entity and returns the logical location of that entity. For instance, *cam-area(o)* is a location function that returns the logical location of a given camera. One location can be related to another location and this

relation is normally determined using a location operator. In the literature, the commonly used location operators include *contains*, *equals* and *overlaps* that check whether a given location contains, is equal to, or overlaps another location, respectively. Let $\mathcal{LOP} = \{contains, equals, overlaps\}$ be the set of these location operators. Using the sets given above, a location constraint in our access control mechanism can be specified as given in *Definition-1*, below.

**Definition-1 [Location constraints].** *Given a set of logical locations $\mathcal{LL}$, a set of physical locations $\mathcal{PL}$, a set of location functions $\mathcal{FL}$, and a set of location operators $\mathcal{LOP}$, a location constraint is defined as follows:*

- $pl_i$ *lop* $ll_i$ , where $pl_i \in \mathcal{PL}$, $ll_i \in \mathcal{LL}$, and $lop \in \mathcal{LOP}$
- $ll_i$ *lop* $ll_j$ , where $ll_i, ll_j \in \mathcal{LL}$, and $lop \in \mathcal{LOP}$
- $loc(x)$ *lop* $ll_i$ , where $loc \in \mathcal{FL}$, $ll_i \in \mathcal{LL}$, and $lop \in \mathcal{LOP}$
- If $lc_1$ and $lc_2$ are location constraints and $lop \in \mathcal{LOP}$, then $lc_1 \wedge lc_2$, $lc_1 \vee lc_2$, and $lc_1$ *lop* $lc_2$ are also location constraints.

#### 5.1.1.2 Temporal Constraints

Temporal information can be either a time instant or a time interval. A time instant is a discrete point on the time line, whereas a time interval is a continuous set of time instances. When a user requests access, the time of request – i.e. a time instant – is evaluated against the temporal constraints defined in the policy. Below we describe how a temporal constraint is formed.

Let $T'$ be the set of time instances, for example, an element of $T'$ is: 2015.07.08.11 : 23 : 56. Whereas time intervals are normally divided into two distinct types: non-recurring and recurring. A *non-recurring* time interval is a range of time which does not repeat, e.g. $2015.06.26 - 2015.07.08$. A *recurring* time interval specifies a range of time which repeats periodically, e.g., the daily recurring interval $09{:}00{-}17{:}00$ repeats itself every day. Based on the repetitive period, recurring intervals are divided into four sub-types [21]: daily, weekly, monthly and yearly. The monthly and yearly intervals may be further divided into subcategories based on the unit time intervals used in each type: day, week, month. A time interval is represented as:

$\{x_1, x_2, ... x_n\}.unit.period$, where $x_i$ represents a unit time interval.

**Daily interval:** A daily interval represents a range of time in a day, denoted by $\Gamma D$, and uses the following format: hours:minutes:seconds − hours:minutes:seconds, e.g. 09:00:00−17:00:00.

**Weekly interval:** A weekly interval represents the days in a week, denoted by $\Gamma W_d$, and is defined as follows:

$$\Gamma W_d = \{x_1, x_2, ... x_n\}.day.week, \text{ where } 1 \leq x_i \leq 7$$

For instance, $\{2, 4, 6\}.day.week$ implies Monday, Wednesday, and Friday every week.

**Monthly interval:** A monthly interval represents the days or weeks in a month during which access is granted. Day interval in a month and week interval in a month are denoted by $\Gamma M_d$ and $\Gamma M_w$ respectively, and are defined as follows:

$$\begin{aligned} \Gamma M_d &= \{x1, x2, ... x_n\}.day.month, \text{ where } 1 \leq x_i \leq 31 \\ \Gamma M_w &= \{x1, x2, ... x_n\}.week.month, \text{ where } 1 \leq x_i \leq 5 \end{aligned}$$

Using the above definitions, $\{1, 15\}.day.month$ represents first and fifteenth days of each month, and $\{2, 3\}.week.month$ represents second and third week of every month.

**Yearly interval:** A yearly interval is used to specify the days, weeks or months in a year during which access is granted. For yearly intervals, we may define day interval in a year ($\Gamma Y_d$), week interval in a year ($\Gamma Y_w$) and month interval in a year ($\Gamma Y_m$) as follows:

$$\begin{aligned} \Gamma Y_d &= \{x1, x2, ... x_n\}.day.year, \text{ where } 1 \leq x_i \leq 366 \\ \Gamma Y_w &= \{x1, x2, ... x_n\}.week.year, \text{ where } 1 \leq x_i \leq 53 \\ \Gamma Y_m &= \{x1, x2, ... x_n\}.month.year, \text{ where } 1 \leq x_i \leq 12 \end{aligned}$$

For example, $\{30, 40\}.week.year$ specifies that access is granted for thirtieth and fortieth weeks of every year.

Let $\Gamma I = \{\Gamma D \cup \Gamma W_d \cup \Gamma M_d \cup \Gamma M_w \cup \Gamma Y_d \cup \Gamma Y_w \cup \Gamma Y_m\}$ represent the set of intervals. In the literature, the commonly used temporal operators include *before*, *after* and *during* that check whether a given temporal term occurs before, after or during another temporal term, respectively. Let $\mathcal{TOP} = \{$*before, after, during*$\}$ be the set of these temporal operators. Using the sets given above, a temporal constraint in our access control mechanism can be specified as given

in *Definition-2*, below.

**Definition-2 [Temporal constraints].** *Given a set of discrete time points $T'$, a set of time intervals $\Gamma I$, and set of temporal operators $\mathcal{TOP}$, a temporal constraint is defined as follows:*

- $t_i$ *top* $t_j$ , where $t_i, t_j \in T'$, and *top* $\in \mathcal{TOP}$
- $t_i$ *top* $t_r$ , where $t_i \in T'$, $t_r \in \Gamma I$, and *top* $\in \mathcal{TOP}$
- $t_i$ *top* $t_j$ , where $t_i, t_j \in \Gamma I$, and *top* $\in \mathcal{TOP}$
- If $tc_1$ and $tc_2$ are temporal constraints and *top* $\in \mathcal{TOP}$, then $tc_1 \wedge tc_2$, $tc_1 \vee tc_2$, and $tc_1$ *top* $tc_2$ are also temporal constraints.

The spatial and temporal constraints, defined above, will now be used in forming resource object expressions and in specifying the conditions, as discussed below.

### 5.1.2 Protected Resource Objects

The objects to be protected by the access control model in video surveillance systems can be divided into two categories: i) live camera feeds, ii) recorded video, collectively referred to as resource objects in this chapter. Each of these categories have attributes associated with them. For instance, the attributes associated with live camera feeds may include *cam-id*, *cam-type*, *cam-area*, and *loc-type*. *Cam-id* is the unique identifier of each camera, *cam-type* defines the type of camera, e.g., pan-tilt-zoom, covert camera, auto-mobile camera etc., *cam-area* specifies the logical location where the camera is deployed in, e.g., Times Square, Brooklyn_east etc., and *loc-type* states the type of location where the camera is deployed, e.g., bus-stop, shopping mall, street etc.

We assume that the videos are stored together with the metadata associated with them. As mentioned earlier, longer videos are normally segmented into smaller video units, called video shots. Each video shot is stored along with the metadata, using solutions such as MPEG-7. The metadata information also includes the starting and ending timestamps of the video shot. Thus, the attributes associated with resource objects also include the annotations extracted from the videos using the video analytics algorithms that extract semantic data contained in the videos. The semantic data extracted from the video includes both the semantic objects, e.g., humans, cars, bicycles etc., and semantic events, e.g., vandalism, fire, bullet-fire etc., contained in the video. Note the difference between the terms semantic objects and resource objects. The term semantic

object represents the objects contained in a video, e.g. vehicle, human etc., whereas resource objects represent the objects – live camera feeds and recorded videos – whose access is to be protected using access control mechanism. It should also be noted that the term objects used earlier in this thesis, particularly in Chapter-4, is semantically similar to the term resource objects. We user the term resource objects, in this chapter, to distinguish between semantic objects and resource objects.

The semantic objects and semantic events extracted from the video depend on the specific application of video surveillance and may range from simple events e.g., motion-detection, crossing a fence to complex events like explosive detection, luggage left behind, nozzle-fire etc. Some of these annotations may be extracted by the camera such as object detection and motion detection, whereas others maybe extracted by an intermediate server when it receives the videos from the cameras. The events may also be reported manually or using physical non-camera sensors deployed near cameras which may report events such as burglary or fire.

Storing the metadata information linked to the videos provides two advantages: i) it enables to retrieve the videos using metadata, ii) this information may be used in specifying the access control policy. It is important to note that the access control policy needs to refer to resource objects based on their attributes, as stated in § 3.4.2 (summarized in Table 5.1), rather than simply using their identifiers. Hence, in the access control policy, the resource objects may be denoted by attributes – including extracted annotations – associated with them. For example, a user may be granted access to all videos from Manhattan area recoded during 10am − 12am which contain a bicycle.

An important consideration in video surveillance systems is that an event or object detected in a video stream may cause to activate an environment attribute and hence may affect the access to further live videos. For example, once a cross-fence event is detected in a specific area, the access privileges of users in this area may be elevated to higher privileges, e.g. a patrolling observer may be given access to unblurred videos in order to recognize and apprehend the culprit.

Below we formalize the resource object attributes which are then used in defining resource object expressions and in specification of conditions.

**Definition-3 [Resource object attributes].** *Let $\mathcal{OIV}$ be the set of variables ranging over the resource object identifiers. Let $\mathcal{OA} = \{oa_1, oa_2, ... oa_n\}$ be the set of resource object attributes. We define these attributes to be either atomic- or set-valued, where an atomic object attribute may hold a single value whereas a set-valued attribute may hold multiple values. Let $\mathcal{OAV} = \{oa_1(v_{11},$*

$v_{12}, \ldots v_{1k})$, $oa_2(v_{21}, v_{22}, \ldots v_{2l}), \ldots$ $oa_n(v_{n1}, v_{n2}, \ldots v_{nm})\}$ *be the set of possible values for each resource object attribute. Let* $\mathcal{OATT}$ *be the set of resource object attribute functions defined for elements of* $\mathcal{OA}$*. Each attribute function takes an element* $x \in \mathcal{OIV}$ *and returns the value(s) of the given attribute. Let* $\mathcal{OP}$ *be a set of operators defined over the attributes.*

Note that the set of operators $\mathcal{OP}$ above includes the spatial and temporal operators defined earlier (*Definition-1* & *Definition-2*) as well as any other semantically relevant operators defined for attributes, in the application.

In the access control policy, we use resource object expressions to represent the set of resource objects which a role is authorized to access. The resource object expressions consist of resource object attributes which may also include the metadata information. The resource object expressions are formed as defined in *Deinition-4*, below.

**Definition-4 [Resource object expression].** *Given the set of variables ranging over resource object identifiers* $\mathcal{OIV}$*, the set of resource object attribute values* $\mathcal{OAV}$*, the set of resource object attribute functions* $\mathcal{OATT}$*, and the set of operators* $\mathcal{OP}$*, a resource object expression is built from atoms which can be defined as follows:*

- $p(x)$, where $x \in \mathcal{OIV}$, and $p \in \mathcal{OATT}$
- $p(x)$ *op* $v$, where $x \in \mathcal{OIV}$, $p \in \mathcal{OATT}$, $v \in \mathcal{OAV}$, and *op* $\in \mathcal{OP}$
- If $roe_1$ and $roe_2$ are resource object expressions and *op* $\in \mathcal{OP}$, then $roe_1 \wedge roe_2$, $roe_1 \vee roe_2$, and $roe_1$ *op* $roe_2$ are also resource object expressions.

*[Examples: Resource object expressions].* The following are examples of resource object expressions:

1. *roe1: {("Brooklyn" contains cam-area(o)) ∧ (timestamp(o) during ([08:00:00 − 16:00:00] ∧ {2,3,4,5,6}.day.week))}.* This is an expression denoting the resource objects from 'Brooklyn' area with timestamp between 8AM to 4PM during weekdays (Monday – Friday).
2. *roe2: (loc-type(o) = "mall") ∧ (timestamp(o) after 2015.03.10.11:00:00) ∧ (timestamp(o) before 2015.03.12.23:59:59) ∧ ("Manhattan" contains cam-area(o)).* This resource object expression denotes those objects which have location type 'mall', belong to the area 'Manhattan', and were recorded during 11PM on March 10, 2015 and 12PM midnight on March 12, 2015.

3. *roe3: (cam-area(o) equals "Brooklyn_east") ∧ (semanticObjects(o) includes "red-car").* This expression refers to the resource objects from the 'Brooklyn_east' area which contain a red-colored car in the sematic objects extracted from the videos.
4. *roe4: (loc-type(o) = "bus-stop") ∧ (semanticEvents(o) includes "fire").* This expression specifies the objects having location type 'bus-stop' with fire-incident detected in the videos.

Note that the above resource object expression examples also include spatial and temporal constraints, as defined in previous section, since the resource objects are associated with spatio-temporal attributes too. This implies that the set of resource object attribute functions $\mathcal{OATT}$ also includes functions related to spatial and temporal attributes which take an object and return either spatial or temporal attribute value for that object. The functions *timestamp()* and *cam-area()* are examples of such attribute functions.

### 5.1.3 Authorized Users

In many applications, identity or role of a user is sufficient to determine whether or not access to a particular object is to be granted. Unlike these applications, in video surveillance, we need to consider other attributes of user too, e.g. location, while evaluating a user's access request. There exist two main types of users in video surveillance: those who need regular access to the data, called regular observers, and those who need to access data only occasionally, called responding observers, as discussed in scenarios (cf. § 3.4.1).

A major difference between regular observers and responding observers is that the former work proactively for serving the purpose of surveillance while the latter work reactively in response to occurrence of an anomalous event. The examples of regular observers include observers in the monitoring room and patrolling observers. Whereas, a patrolling observer responding to an incident, or a fire-man responding to a fire-alarm are examples of responding observers. The important factors in granting access to regular observers are attributes of both user and resource object, as well as environment attributes. The regular observers may be given low privileged access in normal circumstances and high privileged access in case of an emergency. On the other hand, the responding observers are granted occasional access mainly based on occurrence of an incident.

As the proposed access control mechanism is based on the AERBAC model, the role of a user is the core user attribute that determines the set of permis-

sions allowed to the user. Examples of other user attributes, in addition to the role, include user duty timings, user response area, location of a user, etc. The permissions are decided based on user's role and certain other user, object and environment attributes.

**Definition-5 [User attributes].** *Let $\mathcal{UIV}$ be the set variables ranging over user identifiers, let $\mathcal{UA} = \{ua_1, ua_2, ... \ ua_n\}$ be the set of user attributes where each attribute is either atomic- or set-valued. Let $\mathcal{UAV} = \{ua_1(v_{11}, \ v_{12}, ... \ v_{1k}), ua_2(v_{21}, v_{22}, \ldots v_{2l}), ... \ \ ua_n(v_{n1}, v_{n2}, ... \ v_{nm})\}$ be the set of possible values for each element defined in $\mathcal{UA}$. Let $\mathcal{UATT}$ be the set of user attribute functions defined for each element in $\mathcal{UA}$, where each attribute function may take an element $x \in \mathcal{UIV}$, and returns the value(s) of the given attribute.*

As discussed in Chapter-3, the organization responsible for operating and working of the video surveillance systems is named the operator. It is to be noted that it may not be feasible for the operator to assign specific roles and other attributes to all the users in the system, including those from Collaborating Organizations (CO), e.g., police and fire-brigade employees. Doing so would make the task of administration cumbersome as it puts responsibility of reassigning and revoking of attributes for CO users on the operator. A more feasible alternative is that the COs assign attributes to a responding employee when the need arises. The video surveillance operator may inform the COs about an incident and its location and the relevant CO in turn finds its employees near the incident location and assigns them the relevant roles and other required attributes. The responding observers may then access the data from incident location as long as the operator can verify the attributes presented by the responding observers. Security Assertion Markup Language (SAML) [101] may be used for secure exchange of attributes between COs and the video surveillance operator.

### 5.1.4 Environment Attributes

As discussed in § 4.1, attributes that capture the external factors of the situation in which the access takes place are called environment attributes. Current-time, temperature, occurrence of an incident or other information which not only pertains to a specific object or user, but may hold for multiple entities is typically modeled as environment attribute. The state of the environmental attributes may be captured via hardware (e.g. sensors) or software mechanisms, including automatic event detection in videos, that monitor and report changes in the environment.

**Definition-6 [Environment attributes].** *Let $\mathcal{EA} = \{ea_1, ea_2, ... ea_n\}$ be the set of environment attributes where each attribute is either atomic- or set-valued. Let $\mathcal{EAV} = \{ea_1(v_{11}, v_{12}, ... v_{1k}), ea_2(v_{21}, v_{22}, ... v_{2l}), ... ea_n(v_{n1}, v_{n2}, ... v_{nm})\}$ be the set of values for each element defined in $\mathcal{EA}$. Let $\mathcal{LIV}$ be the set variables ranging over location identifiers, and $\mathcal{EATT}$ be the set of environment attribute functions defined for elements of $\mathcal{EA}$, where each attribute function may take either null or an element $x \in \mathcal{LIV}$, and returns value(s) of given attribute.*

The above definition formalizes environment attributes which are now used, along with resource object attributes (*Definition-3*) and user attributes (*Defnition-5*), in specification of conditions.

### 5.1.5 Condition Specification

Being based on AERBAC, a permission in ROAMVS is constrained by one or more conditions, which must be evaluated to be true in order for the user to exercise that permission. Unlike resource object expressions which are formed using only resource object attributes, a condition associated with a permission may contain attributes of all entities i.e. users, resource objects and environment.

**Definition-7 [Condition expression].** *Given the resource object, user and environment attributes as formalized in Definition-3, Definition-5, and Definition-6, respectively, let us suppose $p, g \in \{\mathcal{OATT} \cup \mathcal{UATT} \cup \mathcal{EATT}\}$, $x \in \{\mathcal{OIV} \cup \mathcal{UIV} \cup \mathcal{LIV}\}$, $op \in \mathcal{OP}$, and $v \in \{\mathcal{OAV} \cup \mathcal{UAV} \cup \mathcal{EAV}\}$,then a condition expression is defined as follows:*

- $p(x)$ *op* $v$ is a condition expression
- $p(x)$ *op* $g(x)$ is a condition expression
- If $ce_1$ and $ce_2$ are condition expressions and *op* $\in \mathcal{OP}$, then $ce_1 \wedge ce_2$, $ce_1 \vee ce_2$, and $ce_1$ *op* $ce_2$ are also condition expressions.

*[Examples: Condition expressions].* Examples of condition expressions include:

1. *ce1: userArea(u) contains cam-area(o) ∧ time-of-day() during userDuty(u).* This expression specifies that in order to exercise the associated permission, the camera area must be contained by the current area of the user and the time of access must lie within user's duty timings.

2. *ce2: userAssignedArea(u) equals "Manhattan_ west" ∧ "Manhattan_ - west" ∈ AlarmedRegions().* This condition expression shall be evaluated to be true if user has been assigned to 'Manhattan_ west' and alarm has gone off in that region.

### 5.1.6 Privilege Modes

Access control for video surveillance requires operations that are semantically meaningful for video surveillance data. In order to achieve the goal of maximum utilization of a video surveillance system while protecting the privacy of the individuals recorded, different users may be allowed to access the data with different levels of privacy protection depending on the current contextual information. We achieve this by allowing the users to access video data by specifying the properties of the video and the actions which may be performed on the video data. Properties of the video data may include *frame-rate*, *video resolution*, *privacy protection*, etc. Whereas the actions may include annotations, view, zoom-in, play-back, search and identify. The annotations action allows a user to observe the metadata associated with the videos, such as location, timestamp, semantic objects and events contained in the video, etc. The view action allows to watch the videos. The zoom-in and play-back actions allow zooming into a certain video and playing back a video, respectively. The search action allows to search for an individual, object or a set of video segments, and identify allows to find the identity of an individual in a video, if known by the system, e.g. using remote biometrics [64]. An example lattice of video properties is depicted in Fig. 5.1. In this lattice, we use different possible values for video properties including the frames per second (FPS), the video resolution and the privacy protection property. The privacy protection property that hides the identity revealing regions (e.g. faces) of the objects in video, recorded by video surveillance, may be either silhouettes, blurred or clear. Silhouettes privacy property replaces the semantic objects in the video with dummy figures, blurred privacy property blurs the identity revealing regions, whereas clear privacy property shows the video without protecting the privacy. Several other alternatives of these video properties may be possible.

A privilege mode for the video surveillance data is formed by joining a set of actions with a combination of video properties values. This allows combining together different actions and the properties of the video data, to be assigned as a single operation to a user. Suitable privilege modes should be devised corresponding to video properties and actions performed on video data, as per application requirements. For example, a privilege mode in normal circumstances may allow to access video data with following video properties: 14 FPS, 320x240 resolution and with blurred privacy-sensitive regions, while the actions

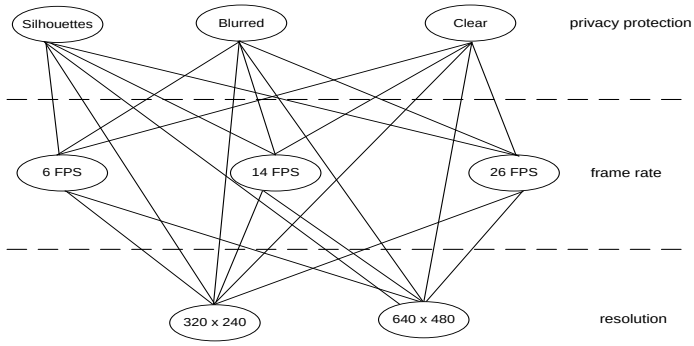


**Figure 5.1:** An example lattice of video properties

allowed may include view, annotations, and playback actions. As part of our mechanism, we devise four different privilege modes which can be applied on video data: low-access, default, high-access and full-access. Table 5.2 shows these privilege modes, each consisting of a unique combination of video properties and allowed actions. Note that the video properties, the privilege modes, and the actions that may be performed on videos given here are not supposed to be exhaustive in nature. For instance, the action to change the direction of a camera is not included in the given privilege modes. Depending on the application requirements, other privilege modes may be defined based on the used actions and video properties.

**Table 5.2:** Privilege modes with video properties and actions

| **Privilege mode** | **Semantics** | |
|---|---|---|
| | **Video properties** | **Actions** |
| low-access | 6 FPS, 320x240, silhouettes | view |
| default | 14 FPS, 320x240, blurred | view, annotations, play-back |
| high-access | 26 FPS, 640x480, clear | view, annotations, play-back, zoom-in |
| full-access | 26 FPS, 640x480, clear | view, annotations, play-back, zoom-in, search, identify |

The privilege modes can have ordering relationships among them, in terms of increasing power, indicating that a privilege mode is subsumed by the other. We represent this ordered privilege mode relationship by $\prec_p$, such that $pm_i \prec_p pm_j$ indicates that privilege mode $pm_j$ subsumes the privilege mode $pm_i$. We define the order of given privilege modes to be: low-access $\prec_p$ default $\prec_p$ high-access $\prec_p$ full-access. Based on changed circumstances, e.g., occurrence of an incident such as fire, the privilege mode allowed to a user may be shifted from low privilege mode (e.g. default) to a high privilege mode (e.g. high-access).

### Example roles and permissions

Now that we have defined the necessary concepts used in our ROAMVS approach, Table 5.3 presents a few example roles along with assigned permissions and their corresponding conditions. As discussed earlier, the permissions comprise of resource object expressions and privilege modes (i.e. operations) and condition associated to a permission must be true in order for the user to exercise that permission. Resource object expressions can only contain resource object attributes comparing them with potential attribute values. In order to perform direct comparisons between resource object attributes and user attributes, conditions are used, which may also use environment attributes. For example, in Table 5.3, the permission in *Room_ observer* role allows a user assuming this role to access all types of cameras (e.g. covert, overt) from all types of locations (e.g. bus-stop, street) with *default* privilege mode. This permission is constrained by a condition which states that the user can exercise this permission when the accessed camera is deployed in the area where user is currently located and the time of access is between 8AM and 4PM.

**Table 5.3:** Example roles with permissions and associated conditions

| **Role** | **Permissions** | **Conditions** |
|---|---|---|
| Room_observer | [(loc-type(o) = all ∧ cam-type(o) = all), default] | [userArea(u) contains cam-area(o) ∧ current-time() during (08:00:00 − 16:00:00)] |
| Patrolling_observer | [(loc-type(o) ∈ {bus-stop, shopping-mall, street} ∧ semantic-objects(o) includes {human, vehicle}), default ] | [userArea(u) contains cam-area(o) ∧ current-time() during (08:00:00 − 16:00:00)] |
| | [(cam-type(o) ∈ {covert, overt} ∧ loc-type(o) ∈ {shopping-mall, street}), high-access] | [userResponseArea(u) contains cam-area(o) ∧ env-mode(cam-area(o)) = alarm] |
| External_observer | [(loc-type(o) = shopping-mall ∧ cam-type(o) = overt), full-access] | [userResponseArea(u) contains cam-area(o) ∧ env-mode(cam-area(o)) = emergency] |

## 5.1.7 Attribute Hierarchy and Derived Permissions

ROAMVS is based on AERBAC which itself is an extension of RBAC; hence users are assigned to roles and roles are assigned permissions. Notice that the permissions in ROAMVS refer to the objects using resource object expressions. Therefore, in ROAMVS, permissions may be implicitly derived from the explicitly specified ones based on hierarchies of resource object attributes, partial order

defined on the set of privileges and obviously due to role hierarchies. Hence there exist three distinct categories of derived permissions: resource object attribute hierarchy, privileges order derivation and role hierarchy. We do not discuss role hierarchies since they are semantically similar to role hierarchies in RBAC i.e. a role $X$ which is senior to a role $Y$ inherits all the permissions assigned to the role $Y$. Below we discuss derived permissions based on partial order among the privilege modes and the resource object attribute hierarchy.

#### 5.1.7.1 Privilege Mode Derivation

As described in the previous section, one privilege mode may subsume another privilege mode. Permissions may, therefore, be derived based on existence of partial order among the privileges defined in the application.

The partial ordering among the privilege modes allows certain permissions to a role which are not explicitly assigned to that role. Suppose a permission $p =< oe_k, m_j >$, where $oe_k$ is a resource object expression and $m_j$ is a privilege mode. Suppose there exists a privilege mode $m_i$, such that $m_i \prec_p m_j$ then the derived permission $p' =< oe_k, m_i >$ is also available to a role having the permission $p$. For instance, the permission specified in *External_ observer* role, in Table 5.3, allows a user in this role to access all resource objects – of type *overt*, and *shopping_ mall* as their location type – with *full-access* privilege mode. Due to the order of privilege modes defined in our mechanism i.e. *low-access* $\prec_p$ *default* $\prec_p$ *high-access* $\prec_p$ *full-access*; a user in *External_ observer* role can also access the resource objects – denoted by the attributes specified in the permission – with *low-access*, *default*, or *high-access* modes, in addition to *full-access* privilege mode.

#### 5.1.7.2 Attribute Hierarchy

Attributes that are used in resource object expressions may also be linked hierarchically. For instance, an attribute representing the location of a resource object may have a semantic hierarchy, e.g., Brooklyn_east is a sub-region of Brooklyn which is in turn a sub-region of New York city, as shown in Fig. 5.2. Similarly, there may exist hierarchies between other resource object attributes including the semantic object and events contained in the videos. An example of the semantic object hierarchy is shown in Fig. 5.3. An example hierarchy among semantic events is given in Fig. 5.4. Note that these figures provide example hierarchies and may differ from application to application. Based on the attribute hierarchy, a permission that allows to access resource objects as-

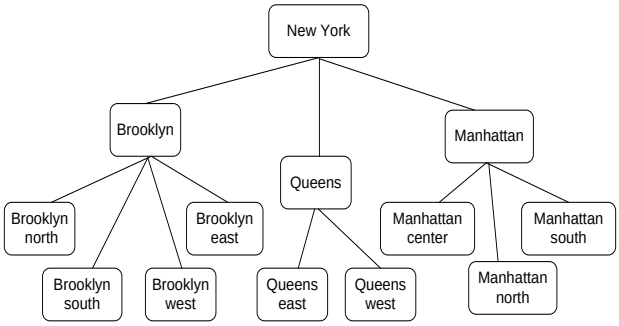


**Figure 5.2:** Example location hierarchy

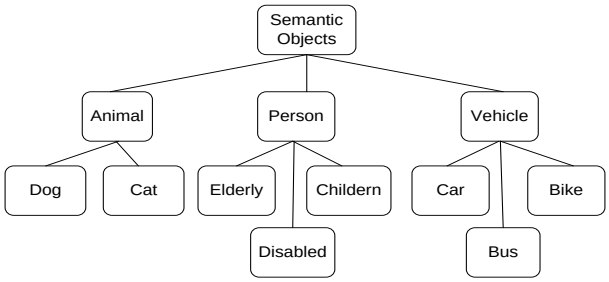


**Figure 5.3:** Semantic object hierarchy example

sociated with New York region allows derivation of additional permissions for resource objects associated with more specific resource object location, in this case resource objects associated with Manhattan region, as per the example hierarchy defined in Fig. 5.2.

It must be noted that a resource object expression normally contains multiple object attributes hence a derived permission allows to access only those resource objects which are represented by a more specific object attribute but further constrained by additional object attributes which exist in that resource object expression.

We represent hierarchy of object attributes using: $oa_x \prec_{oa} oa_y$ which represents that object attribute $oa_x$ is more specific than $oa_y$. Due to the existence of attribute hierarchy, a resource object expression may be a sub-expression of another resource object expression which consists of resource object attributes higher in the hierarchy. We represent this relationship as $oe' \prec_{oe} oe$ which denotes that resource object expression $oe'$ is a sub-expression of $oe$. Suppose there exists a permission $p =< oe, m_i >$ where resource object expression $oe$ consists of object attributes $oa_i$, $oa_j$ and $oa_k$. Suppose there exists another resource object expression $oe'$ which consists of object attributes $oa_i$ , $oa_j$, and

$oa_l$, such that $oa_l \prec_{oa} oa_k$, then $oe'$ is a sub-expression of $oe$. Hence the permission $p' =< oe', m_i >$ is a derived permission from $p$ and is also available to a role which has been assigned the permission $p$.

Due to the derived permissions, our approach allows a larger set of permissions associated to a role comprised of relatively few permissions explicitly assigned to that role.

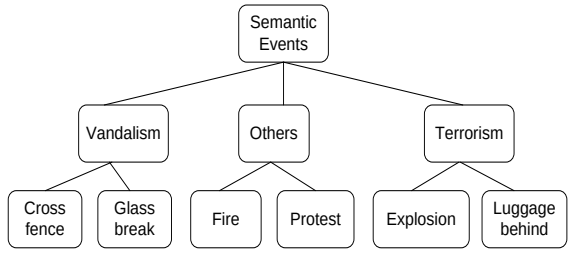


**Figure 5.4:** Semantic event hierarchy example

## 5.2 Access Control Decisions

Access control deals with verifying whether a user requesting access to a certain resource object using a specific privilege can be granted access, according to the access control policy. Note that, in ROAMVS, the access control policy is specified in a role-oriented fashion. Each role is assigned a set of permissions which may be exercised by a user assuming that role, provided that the condition associated with the requested permission is fulfilled. Each permission comprises of resource object expression and privilege mode. In order to process an access request, we do not consult the whole authorization policy base, rather only the permissions available in the user's session are evaluated.

In video surveillance systems, the resource object to be accessed may be either live camera feeds or videos recorded earlier. An important consideration in video surveillance environments is that the user's request may also be based on the attributes of the resource objects. For instance, a user might want to access all live feeds with specified characteristics e.g., cameras with *loc-type* = *street* and *cam-area* = *Brooklyn_ west*. Hence, the authorization process should consider both the possibility of an object identifier or object attributes in the request. As discussed in previous chapter, AERBAC allows to specify an access request using a resource object identifier as well as resource object attributes (cf. § 4.2.1). We also discussed the algorithms for evaluation of identifier-based and attribute-based requests. Identifier-based request is evaluated in the same

manner as given in previous chapter. However, attribute-based requests are evaluated slightly differently, due to the nature of resource objects and the permission derivation.

Specifying the object attributes in the request implies that the user wishes to access all those resource objects which have the specified attribute values. As mentioned above, a user may be interested in accessing either live feeds or recorded videos. We may distinguish between requests for live feeds and recordings based on attributes specified in the request. For instance, if a request specifies the timestamp for which the data is required, it implies that the user is interested to access recordings. On the other hand, if no timestamp is mentioned in the request, the user rather requires live feeds.

In Chapter-4, we discussed two algorithms for attribute-based request evaluations: i) resource query evaluation, ii) attribute-values evaluation. Below we provide slightly modified versions of these algorithms, distinguishing between live and recorded videos and allowing derivation of permissions, to show how user requests will be evaluated using ROAMVS in the video surveillance environments.

**Resource Query Evaluation**

In this approach, the user request contains an expression that denotes a set of resource objects to be accessed. The access request can be represented as a triple: $Req =< se, re, m >$, where $se$ is the session identifier, $re$ is the request expression, and $m$ states the requested privilege. An example user request could be: $Req =<se,\ (cam\text{-}type = ptz \wedge loc\text{-}type = street \wedge cam\text{-}area = Brooklyn),\ default>$ which states that the owner of the session $se$ wishes to exercise the *default* privilege on the resource objects denoted by the given request expression. The request evaluation steps in this approach are given by the algorithm in Fig. 5.5.

The algorithm receives as input the access request $Req$ and returns all the resource objects that are relevant to the request expression and that owner of session $se$ is allowed to access according to the permissions in session $se$. The given expression is examined, by *liveOrRec()* function, to determine whether the user intends to access live feeds or recordings. This is done by checking the resource object attributes mentioned in the request. Once determined, the $searchObjects()$ function converts the expression into a query and retrieves the resource objects, represented by the given expression, from the resource objects database. Next step is to find the applicable resource object expressions by matching the user's requested privilege with the ones mentioned in the permission set existing in the user's session, and by verifying whether the resource object expression is relevant to the requested object category (live feeds or

**Algorithm 3**

**Input:** An access request: Req = <se, re, m >comprising of session identifier *se*, request expression *re*, and privilege mode *m*.

**Output:** 1) Grant and return authorized resource objects, 2) Deny otherwise

```
1: relevant_expressions = φ;
2: object_set = φ;
3: authorized_objects = φ;
4: obj_category = liveOrRec*(re);
5: object_set = searchObjects**(re, obj_category);
6: if object_set ≠ φ then
7:     for all perm<res_object_exp, op>∈ avail_session_perms do
8:         if m ⪯_p pm AND isRelevant†(res_object_exp, obj_category) then
9:             relevant_expressions ← relevant_expressions ∪ res_object_exp;
10:        end if
11:    end for
12:    for all object ∈ object_set do
13:        for all res_object_exp ∈ relevant_expressions do
14:            if evaluate††(res_object_exp, object) then
15:                if evalCond‡(condition, object, session_user(se)) then
16:                    authorized_objects ← authorized_objects ∪ object;
17:                    break;
18:                end if
19:            end if
20:        end for
21:    end for
22: end if
23: if authorized_object ≠ φ then
24:    return authorized_objects;
25: end if
26: return Reject;
```

*liveOrRec(re) receives the request expression *re* and determines whether user is interested to access live feeds or recorded videos. It returns the category (live feeds or recordings) that user intends to access.

**searchObjects(re, obj_category) returns a set of resource objects existing in the resource database that are denoted by the constraints specified in expression *re*, in the request.

†isRelevant(res_object_exp, obj_category) returns TRUE if value of *obj_category* is "live" and *res_object_exp* comprises of camera attributes, or if value of obj_category is "recordings" and *res_object_exp* comprises of recorded video attributes, and FALSE otherwise.

††evaluate(res_object_exp, object) returns TRUE if *res_object_exp* evaluates to true for the given *object*, else returns FALSE.

‡evalCond(condition, object, session_user(se)) returns TRUE if given *condition* evaluates to true for the given *object* attributes and the attributes of the user and the environment.

**Figure 5.5:** Evaluation of attribute-based request using query result evaluation

recordings). Once the resource object expressions are shortlisted, they are evaluated, using the function *evaluate*(), one-by-one for each resource object returned by the query. If a resource object expression and its corresponding condition evaluate to true for an object, the object is added into the list of *authorized_ objects* to be granted to the user. Finally, user is granted access to all the objects in this list.

As discussed above, the resource object expressions are to be evaluated for each returned object, therefore, this approach may prove to be expensive in cases where several resource objects are returned by the query formed based on user's request. A possible variation in attribute-based request processing may be to provide a list of representative thumbnails for the resource objects returned against the user request. The user can then select one of the returned objects and authorization policy is consulted for the selected object, or the user submits a more detailed request based on the intermediate result. However, the information returned, e.g., where and how many cameras are deployed in a region, may itself be classified, and hence may also need authorization before such information can be returned to a user. The next approach, attribute-values evaluation, addresses these issues.

**Attribute-values Evaluation**

In attribute-values evaluation, user's request is evaluated against the resource object expressions, existing in the user's session, before retrieving the actual objects from the resource objects database. In this approach, rather than providing an expression, user specifies his/her access request by specifying the attribute values of the desired resource objects. The user request comprises of three elements: $Req = <se,\ object\_attribute\_values,\ m>$, where *se* is the session identifier, *object_ attribute_ values* specifies the attributes of the requested resource objects, and *m* states the requested privilege. The algorithm receives as input the user request *Req* and returns the resource objects denoted by object attribute values given in *Req*, if request is granted, otherwise the request is denied. An example user request could be: $Req_i = <se,\ (cam\text{-}type = covert;\ loc\text{-}type = mall;\ cam\text{-}area = Manhattan\_center),\ high\text{-}access>$. To process the user request, all those resource object expressions existing in the user's session are identified in which: i) the privilege mode specified in the permission matches with requested privilege, ii) the resource object expression is relevant to the requested object category (live feeds or recordings), iii) the resource object expression uses only the attributes mentioned in the user's request (*attribute_ relevancy()*). Resource object expressions that include an attribute not specified by the user request are not relevant. After shortlisting the resource object expressions, the attribute functions in each resource object expression is given the object attribute values from the user request. Suppose we find a resource object expression: $(loc\text{-}type(o) = mall \wedge cam\text{-}area(o) = Manhattan\_center)$ relevant

**Algorithm 4**

**Input:** An access request: Req = < se, obj_attribute_values, m> comprising of session identifier *se*, object attribute values *obj_attribute_values*, and privilege mode *m*.
**Output:** 1) Grant and return authorized resource objects, 2) Reject otherwise
**Begin:**

```
1: authorized_objects = φ;
2: obj_category = liveOrRec*(obj_attribute_values);
3: for all perm < res_object_exp, op > ∈ avail_session_perms do
4:     if m ⪯_p pm ∧ attributeRelevancy**(res_object_exp, obj_attribute_-
   values, obj_category) then
5:         if evaluate† (res_object_exp, obj_attribute_values) then
6:             if evalCond‡(condition, obj_attribute_values, session_user(se)
   then
7:                 authorized_objects = get_objects‡‡(obj_attribute_values);
8:                 break;
9:             end if
10:        end if
11:    end if
12: end for
13: if authorized_object ≠ φ then
14:    return authorized_objects
15: end if
16: return (Reject)
```

**End**

*liveOrRec(re) receives the request expression *re* and determines whether user is interested to access live feeds or recorded videos. It returns the category (live feeds or recordings) that user intends to access.

**attributeRelevancy(res_object_exp, obj_attribute_values, obj_category) returns TRUE if the given *res_object_exp* uses only those object attribute functions referred in *obj_attribute_values* and which belong to same category as *obj_category*

†evaluate(res_object_exp, obj_attribute_values) returns TRUE if the given *res_object_exp* evaluates to true when the object attribute functions are replaced with *obj_attribute_values*

‡eval_cond(condition, obj_attribute_values, session_user(se)) returns TRUE if the given *condition* evaluates to true for the given object attributes and the attributes of the user and environment

‡‡get_objects(obj_attribute_values) returns a set of resource objects existing in the resource objects database that satisfy *obj_attribute_values*

**Figure 5.6:** Evaluation of attribute-based request using attribute-values evaluation approach

to the user request $Req_i$, given above. Upon picking the values of the object attribute functions *loc-type* and *cam-area* from user given attribute values we get: *(mall = mall ∧ Manhattan_ center = Manhattan_ center)* which would evaluate to true. As soon as a resource object expression and its corresponding condition return true, the user's request is granted and rest of the resource object expressions are ignored.

When an expression returns true, the *get_ objects()* function forms a query based on the attribute values specified in the user request and the user is granted access to all those resource objects returned by the query. This query may restrict the list of returned objects based on any additional attributes mentioned in the user's request. In the example above, the returned result is restricted based on additional object attribute *cam-type* which is mentioned in the user's request but does not exist in the resource object expression which enables the request. The algorithm for this approach is given in Fig. 5.6.

### 5.2.1 Enforcement Architecture

A simplified architecture to enforce the proposed mechanism is shown in Fig. 5.7. The video data is captured by the deployed cameras at their respective locations. The video server is responsible to manage cameras and receive videos from cameras and acts as an interface among cameras, storage server and the access control module. The annotations extracted from the video data are stored, linked to the videos, by the storage server. The videos may be accessed by users, wishing to view the live or recorded videos of a desired location, i.e. live video feeds are often viewed in a special monitoring room, and these live or recorded videos may also be viewed on hand-held devices or a workstation. The reference monitor receives a user request and evaluates the request as per user's session and the format of the access request. The context manager is responsible to monitor and store the contextual information related to users, resource objects and the environment.

Once the criteria in the authorization policy are fulfilled, the retrieved videos are post-processed by the video filtering module before delivering to the user. Post-processing is done to enable different levels of access to the data according to the user authorizations as mentioned in the policy. Post-processing may include, for example, hiding the privacy sensitive regions in the live feeds or recorded videos when showing them to a user, and removing video segments that the user is not authorized to access from the requested time-period, etc. The access levels are determined based on attributes of the user, environment, and resource object (including the semantic objects and events extracted as annotations from the video).

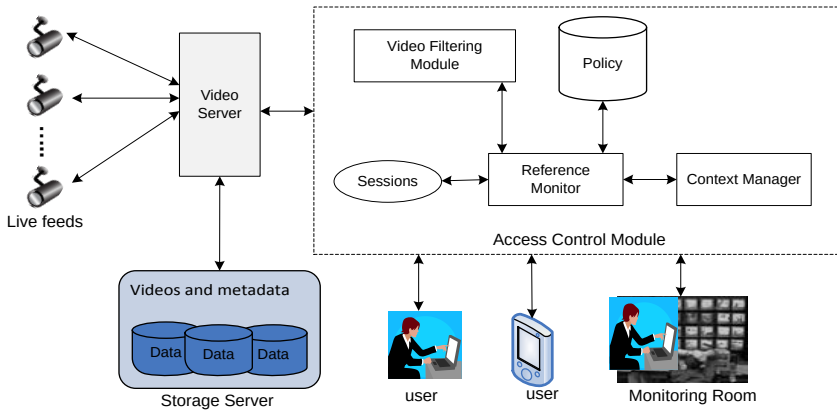


**Figure 5.7:** System architecture for controlling access in video surveillance

### Continuous Enforcement

Continuous enforcement of access control [108] ensures whether the contextual attributes involved in a permission and its corresponding condition continuously hold the same value, during permission execution, which they had at the time of granting a requested permission. For instance, the logical location of a user may change from one area to another. Another example is change in an environment attribute (e.g. emergency) from active to de-active or vice-versa. Different mechanisms to re-evaluate a granted permission have been discussed in the literature [78]. One possibility is to re-evaluate a granted permission periodically after specified intervals of time [108]. Another possibility is to re-evaluate whenever a change in a contextual attribute is detected [65]. It is important to note that not every change in contextual attribute should trigger re-evaluation of the permission. Rather it should be re-evaluated only if the change causes modification in the semantic value of an attribute. For instance, the physical location of a user may change very frequently but the logical location would change only when the user moves out of the boundary of a specific region. Thus, in this case, the re-evaluation should be triggered only when the user's logical location is changed. The context manager needs to keep track of the updated contextual information by monitoring the attributes that may change during the active session of a user. When such a change is detected, the context manager reports to the reference monitor about the change so that the policy may be re-evaluated. Our approach facilitates such frequent checking of ongoing updates in the attributes by re-evaluating the permissions in the user's session, based on his/her roles activated in the session. The proposed mechanism may significantly reduce the number of authorization rules to be evaluated since we do not evaluate the whole policy as opposed to the earlier approaches [23, 24].

## 5.3 XACML Profile and Implementation

The eXtensibel Access Control Markup Language (XACML) [102] is the most widely used policy specification language for access control. It is standardized by the Organization for Advancement of Structured Information Systems (OASIS). In this section, we present an XACML profile for the access control mechanism we proposed for video surveillance systems. We have also implemented a prototype system to demonstrate the feasibility of the proposed approach.

XACML has three major constructs: *policy-set*, *policy*, and *rule*. A *rule* is defined by the triple: *target*, *effect* and *condition*. *Target* in a *rule* restricts the circumstances where the *rule* is applicable, *effect* specifies the outcome of the decision (e.g. permit, deny), and the *condition* is a boolean expression on the set of attributes which has to be satisfied for applying the rule's effect. A *policy* consists of a *target* and a set of rules. The *target* in a policy states the circumstances when the policy is applicable. A *policy-set* is a larger construct and contains a set of policies.

The core and hierarchical XACML profile of RBAC [100] defines a profile for the use of XACML to meet the requirements of RBAC. Because the underlying model, AERBAC, we used for access control in video surveillance, is a role-oriented model; our profile is also role-oriented. The roles are special attributes in our access control mechanism and are assigned permissions, similar to RBAC. We use two generic XACML policies: a Permission <*PolicySet*> and a Role <*PolicySet*>. Permissions are expressed in a permission <*PolicySet*> which defines a set of permissions assigned to a role. Recall that, in our access control mechanism, a permission consists of a resource object expression and a privilege mode, and is constrained by a condition that has to be true in order for the user to exercise that permission. Resource object expressions are formed using resource object attributes whereas conditions may use object, user and environment attributes.

In XACML, we specify the permissions and conditions as a rule using both <*Target*> and <*Condition*> elements. Conditions on the permissions are specified using <*Condition*> element. A permission <*PolicySet*> may also contain <*PolicySetIdReference*> to another permission <*PolicySet*> in order to inherit permissions from another role. Note that direct comparison of two attributes in a <*Target*> is currently not supported by the XACML. It supports comparing an attribute with an attribute value but does not allow comparing two attributes, e.g. *userDepartment* == *resourceDepartment* cannot be compared in <*Target*> element. Such comparisons can be performed only using a <*Condition*>.

Role <*PolicySet*> binds the holder of a specified role to the permissions as-

signed to that role by pointing to a permission <*PolicySet*> using the reference <*PolicySetIdReference*>. Like RBAC, the policies should be stored in the policy repository in such a way that the role <*PolicySet*> instances are always used as initial policy while the permission <*PolicySet*> instances should only be reachable via the corresponding role <*PolicySet*>. This ensures that only subjects having a specific role can access the permissions assigned to that role, specified through permission <*PolicySet*>.

### 5.3.1 Prototype Implementation

Together with the language, the OASIS standard [102] also defines architectural components for enforcing XACML policies. A core component of this architecture is the Policy Decision Point (PDP) which evaluates an access control request against the set of XACML policies and returns the decision. There exist multiple open-source implementations of XACML PDP: Sun's XACML [138], XACML Light [2], XACML Enterprise [3], and Balana XACML [157]. Among these, we chose Balana XACML implementation because of its modular and easily extendible architecture. It is based on Sun's XACML implementation and is the only open-source implementation that supports the latest specification of XACML i.e. XACMLv3.0.

In order to use logical location hierarchy, timestamps in the videos and partial order defined over the privilege modes in the access control policy, we extended XACML and defined following functions:

- *mvaas:functions:location-contained-by* takes two logical locations as input and returns true if the first logical location is contained by the second logical location, according to the location hierarchy defined over the list of logical locations relevant to the application.

- *mvaas:functions:mode-equal-or-superior* takes two privilege modes as input and returns true if the first privilege mode is equal or superior in order than the second privilege mode, according to the partial order defined over the available privilege modes.

- *mvaas:functions:timestamp-interval-allowed* takes four arguments as input and returns true if the timestamp-interval represented by the first two arguments is within the timestamp-interval represented by the last two arguments.

- *mvaas:functions:time-in-range* has been defined as an additional function in Sun's XACML implementation. It takes as input three time values and returns true if the first time value falls in between the second and third time values.

- *mvaas:functions:day-in-given-weekdays* takes two arguments as input and returns true if the first argument value (day) falls in the range of days specified, as formalized in § 5.1.1. We achieve this by converting the current date to day of the week and then verify whether this day of the week falls in one of the given days.

Figure 5.8 shows the role and permission policy-sets for the *Patrolling_ observer* role using a reduced form of XACML: the presentation is a simplified version of XACML as the namespaces are removed, and the attributes, functions and algorithms names are shortened. The example policy allows a subject assuming *Patrolling_ observer* role to access all camera feeds of *loc-type = street*, and *cam-type = point-tilt-zoom* with *high-access* mode, provided that the camera exists in the region where the user is currently located and access time is between 09:00 and 17:00 hrs during weekdays (Monday – Friday).

```
<!-- Role policy set-->
<PolicySet PolicySetId="RPS:patrolling_observer:role" PolicyCombiningAlgId="
    permit-overrides">
    <Target>
      <AnyOf>
        <AllOf>
          <Match MatchId="string-equal">
            <AttributeValue DataType="string">patrolling_observer</
                AttributeValue>
            <AttributeDesignator Category="access-subject" AttributeId="
                subject:role"
            DataType="string"/>
          </Match>
        </AllOf>
      </AnyOf>
    </Target>
 <!-- Use permissions associated with the patrolling_observer role -->
    <PolicySetIdReference>PPS:patrolling_observer:role</PolicySetIdReference>
 </PolicySet>
 <!-- ================================== -->
<!-- Permission policy set-->
<PolicySet PolicySetId="PPS:patrolling_observer:role" PolicyCombiningAlgId="
    permit-overrides">
    <Target/>
<!-- Permissions specifically for the patrolling_observer role -->
    <Policy PolicyId="Permissions:specifically:for:the:patrolling_observer:
        role" RuleCombiningAlgId="permit-overrides">
      <Target/>
 <!-- Permission to view feeds from cameras of location-type = street AND
     camera-type = pan-tilt-zoom with high  access mode; in case the camera
     lies in the same area as of user and access time is within subject's
     duty-timings-->
      <Rule RuleId="view-alarm-mode" Effect="Permit">
        <Target>
          <AnyOf> <AllOf>
            <Match MatchId="string-equal">
              <AttributeValue DataType="string">street</AttributeValue>
              <AttributeDesignator Category="resource" AttributeId="object:
                  location-type" DataType="string"/>
            </Match>

            <Match MatchId="string-equal">
              <AttributeValue DataType="string">pan-tilt-zoom</AttributeValue
```

```
                    >
35                <AttributeDesignator Category="resource" AttributeId="object:
                      camera-type" DataType="string"/>
36              </Match> </AllOf> </AnyOf>
37            <AnyOf> <AllOf>
38              <Match MatchId="function:mode-equal-or-superior">
39                <AttributeValue DataType="string">high-access</AttributeValue>
40                <AttributeDesignator Category="action" AttributeId="action-id"
                      DataType="string"/>
41              </Match> </AllOf> </AnyOf>
42          </Target>
43  <!-- Object's area must be contained by subject's area AND current time must
        fall in the user's duty timings AND specified week days-->
44          <Condition>
45            <Apply FunctionId="and">
46              <Apply FunctionId="function:location:contained-by">
47                <Apply FunctionId="string-one-and-only">
48                  <AttributeDesignator DataType="string" AttributeId="object:
                        area" Category="resource"/>
49                </Apply>
50              <Apply FunctionId="string-one-and-only">
51                <AttributeDesignator DataType="string" AttributeId="subject:
                      current:area" Category="access-subject"/>
52              </Apply>
53            </Apply>
54            <Apply FunctionId="function:time-in-range">
55              <Apply FunctionId="time-one-and-only">
56                <AttributeDesignator Category="environment" AttributeId="
                      current-time" DataType="time"/>
57              </Apply>
58              <AttributeValue DataType="time">09:00:00</AttributeValue>
59              <AttributeValue DataType="time">17:00:00</AttributeValue>
60            </Apply>
61           <Apply FunctionId="function:day-in-given-weekdays">
62              <Apply FunctionId="date-one-and-only">
63                <AttributeDesignator Category="environment" AttributeId="
                      current-date" DataType="date"/>
64             </Apply>
65              <AttributeValue DataType="string">{2,3,4,5,6}.day.week </
                    AttributeValue>
66              </Apply> </Apply>
67          </Condition>
68        </Rule>
69      </Policy>
70  </PolicySet>
```

**Figure 5.8:** XACML policy for the *Patrolling_ observer* role

We extended Balana XACML implementation with context manager which is responsible for keeping record of roles and other attributes of user, resource object and environment. When a request is evaluated by the PDP, the required attribute values are retrieved with the help of Attribute Finder Modules (AFM). We extend Balana XACML implementation with following AFMs: i) role attribute finder module which finds the currently active roles of a user, ii) user attribute finder module which finds the value of a given user attribute for a given user, iii) environment attribute finder module which finds an environment attribute active at a given location, and iv) resource object attribute finder module which finds the value of a given attribute for a given resource object.

This prototype demonstrates the implementation feasibility of the proposed access control mechanism. Video filtering module, given in Fig. 5.7, is orthogonal to this prototype and may be added to post-process the retrieved videos in order to provide different levels of information in a video to different users.

## 5.4 Comparison with Related Work

This section compares the existing solutions relevant to access control in video surveillance systems with the mechanism that we proposed. As discussed in § 3.3.3, very few research efforts have focused on the challenge of access control in video surveillance. However, a reasonable amount of work has been done in related disciplines including multimedia applications and satellite data that defines access authorizations using contents of the data itself. Below we discuss few such prominent efforts and compare them with the proposed mechanism. Moreover, we explain why the existing access control approaches in the area of video surveillance as well as multimedia and satellite applications do not provide the features desired for access control in video surveillance (cf. § 3.4.2, Table 5.1).

***Video Surveillance:*** In the following, we describe how our mechanism is different than the existing approaches which discuss access control in video surveillance. These approaches have already been discussed in the thesis, for more details see § 3.3.3.

Senior et al. [130] present the idea of using multiple privacy levels in video surveillance systems where different users are to be provided different levels of information. The authors suggest using a privacy-preserving console manager that extracts the information components from videos as per the authorization level of the user. Birnstill & Pretschner [25] propose to use two different operational modes called default and alarm. The default mode aims to hide the privacy sensitive regions and shows only the site map view of the surveillance area with type and location of objects. The alarm mode, when activated, shows the video streams without hiding the privacy-sensitive regions. However, in both these solutions [130, 25], the authors neither define an access control model nor do they discuss the structure and language that may be used to specify the authorization policy. They assume that the access control policy is already in place. Moreover, our approach is different than both these solutions [130, 25], in terms of focusing on designing of an access control mechanism whereas these solutions focus on computer vision and pattern recognition algorithms to hide the privacy sensitive regions in a video.

Moncrieff et al. [87] identify the challenge of utilizing the video surveillance system by exposing sufficient need-specific data to the users while preserving

the privacy of people. The authors suggest to devise a dynamic access control mechanism by incorporating the context of the requester in the access control process. However, this paper also does not provide an access control mechanism. The main contribution of this paper is to identify the challenge of a dynamic access control in video surveillance while leaving the designing of dynamic access control model as a goal to be achieved in future research. We take on this challenge of designing a dynamic access control model and propose a mechanism that incorporates contextual information in decision making and preserves privacy of people without under-utilizing the efficacy of the system.

To the best of our knowledge, the only solution discussing the access control model for video surveillance systems is provided by Thuraisingham et al. [145]. This solution makes use of metadata extracted from video data. However, in this solution, the access privileges assigned to a user are based solely on the credentials presented by the subject and no environmental information is considered in the access control policy. Moreover, these credentials are static in nature and are assumed not to change once access has been granted. Furthermore, this solution uses the credential-based (i.e. attribute-based) approach which does not facilitate reviewing permissions assigned to a user and makes it hard to change the access control policy (cf. § 4.3.2). Due to lack of support for reviewing permissions, this model does not offer simplified auditing and policy modification visualization features described in § 3.4.2 (summarized in Table 5.1). These features are typically supported by a role-based approach hence we used the well-known notion of roles in our proposed solution to fulfill these requirements.

***Multimedia applications:*** Various solutions addressing the access control issue in multimedia applications have been proposed. Kodali et al. [72] develop a generalized security framework which enforces access control using already configured access control models. As a multimedia object can be decomposed into several layers of information, it is important that the access to such objects may be controlled based on the contents of these objects. However, this model does not allow specification of access control policy based on contents of the multimedia objects. The most influential access control model for multimedia applications is presented by Bertino et al. [24], where they propose an access control model for video database systems. They define a video stream as a series of frames that describes the semantic context and is associated with annotation. Considering the issues associated with prior access control mechanisms that specify authorizations in terms of user identities or user groups and object identifiers, their solution allows specification of authorization rules using credentials (attributes) to specify users and content expressions to specify objects. Their solution offers varying granularity of protected objects ranging from an entire video to part of a video. The concept of resource object expression in our approach is similar to content-expression in this paper. However, a content expression does not allow referring objects with spatio-temporal infor-

mation. Subsequent research [23] attempts to extend this work by suggesting storage of the videos in a hierarchical fashion with respect to a concept hierarchy, specific to the domain of addressed problem. For instance the video database may contain videos about medical or news domain, thus a hierarchy would be defined for each domain and each video stream would be stored as an element in the given hierarchy. Such a hierarchy is then exploited in the access control policy leading to reduced number of authorization rules exploiting the hierarchical structure of the video database. Being well suited for multimedia based video applications, the video data, in these solutions [24] [23], is not associated with the spatio-temporal characteristics. Besides, these solutions use the concept of restricted objects in order to allow multilayer access to a video. For instance, a user may be allowed to view a video but the faces of people in video are specified as restricted objects. The concept of restricted objects may lead to conflicts in the policy where one rule may allow a certain action whereas another rule may restrict it. Though a conflict resolution mechanism is offered to address this issue, however, such mechanism needs to be consulted each time a new authorization rule is inserted in the policy. To avoid policy conflicts, we rather use privilege modes as a combination of video properties and actions that may be performed on the videos. For example, a privilege mode may allow users to view videos but faces of people or number plates of vehicles may be blurred by specifying video properties in the privilege mode, rather than using negative authorizations. Thus our access control model may achieve the same results without using negative permissions and hence avoiding conflicts. Moreover, these solutions do not consider environmental information, e.g. occurrence of an incident, in the access control policy. Furthermore, because of not being a role-based approach, these solutions [24, 23] do not offer auditability and modification visualization.

***Satellite data:*** Similar to our target domain, satellite images have also the unique property of being associated with specific real-world geographic coordinates and a timestamp that indicates the time when the image was captured. Access control model, proposed by Atluri et al. [12], discusses an access control model for satellite images. The model utilizes spatio-temporal information associated with satellite data by defining spatial and temporal terms and uses the concept of user credentials in order to specify the users in the access control policy. The authors use hierarchical taxonomies both for user credentials and privileges to reduce the explicit specification of authorization rules in the policy. Privilege modes in our solution are semantically different than privileges used in this solution as we form privilege modes by combining actions with video properties. Besides, in applications where attributes may be associated with both users and objects, access control policy may need to check whether there exists a relation between a user attribute and an object attribute. For example, a user associated with Brooklyn-east region may be allowed to access data related to Brooklyn-east region only. Allowing direct comparisons between attributes

**Table 5.4:** Summary of existing access control models relevant to video surveillance

| Features/ Approaches | Metadata-based permissions | User context | Environmental information | Dynamic attributes | Simplified auditing | Modification visualization |
|---|---|---|---|---|---|---|
| **Video Surveillance** | | | | | | |
| Thuraisingham et al. [145] | YES | YES | NO | YES | NO | NO |
| Senior et al. [130], Moncrieff et al. [87], Birnstill et al. [25] | N/A | N/A | N/A | N/A | N/A | N/A |
| **Multimedia Applications** | | | | | | |
| Bertino et al. [24] | YES | YES | NO | NO | NO | NO |
| Bertino et al. [23] | YES | YES | NO | NO | NO | NO |
| Kodali et al. [72] | NO | YES | NO | NO | YES | YES |
| **Satellite data** | | | | | | |
| Atluri et al. [12] | YES | YES | NO | YES | NO | NO |
| Atluri et al. [13] | YES | YES | NO | YES | NO | NO |
| **Our approach** | | | | | | |
| ROAMVS | YES | YES | YES | YES | YES | YES |

of users and objects can significantly reduce the number of authorization rules because authorization rules may be formed without having to use values of certain attributes in the policy. However, none of the above solutions [12, 23, 24] allows performing direct comparisons between attributes of different entities. In a successive research, the work by Atluri et al. [12] is further extended with roles [13]. However, the roles do not encapsulate permissions in this solution rather a role is used as simply an attribute. Hence, auditing the permissions assigned to a user and seeing the impact of a policy change are not supported by these solutions [12, 13] either, due to not being role-based approaches. Moreover, these solutions do not take into account the environmental information, e.g. occurrence of an incident, in making an access control decision.

Table 5.4 provides a summary of the access control models we discussed above with respect to the features desired in access control for video surveillance systems, described in § 3.4.2 (summarized in Table 5.1). Under the video surveillance category, we compare our solution against the one given by Thuraisingham

et al. [145], since this is the only solution which discusses a formal access control mechanism for video surveillance, as mentioned earlier. A yes in a table cell means that the given feature is supported by the approach while no against a feature implies that the requirement is not supported by the solution. Note, however, that no against a feature does not necessarily mean that the given feature cannot be supported by the discussed approach, it rather means that supporting the feature requires more work and in some cases may make the approach prohibitively complex. For instance, supporting the auditability feature using credential-based approach is considered hard to achieve because reviewing the policy for a user requires analyzing the whole policy.

## 5.5 Summary

This chapter presented an access control mechanism for video surveillance systems using the AERBAC approach. As the videos are linked to a particular location and certain time, the formation of spatial and temporal constraints was discussed to use them in the access control policy. The characteristics of the objects to be protected and the users were discussed and it was shown how the permissions can be created using the resource object expressions and the conditions in a role-centered manner. The chapter introduced the privilege modes as a combination of video properties and actions that can be allowed on the video data. The derivation of additional permissions from the explicitly stated permissions was discussed. We then demonstrated that the access control policies following our ROAMVS approach can be specified using XACML policy language. Finally, a comparison of our approach with other existing approaches was presented.

Chapter 6

# Conclusions and Future Directions

Modern video surveillance systems offer advanced functionalities such as automatically detecting and tracking an object in the captured videos. The observers may watch the captured videos not only in physically-controlled monitoring rooms but may also access them ubiquitously over hand-held devices. Pervasive use of such systems motivates the need for controlling access to the videos, in order to avoid performing voyeurism and profiling an individual by those monitoring the videos, thus protecting the privacy of people. In this context, a multilevel dynamic access control mechanism was presented that provides access to different levels of information to different users taking the contextual information into account. This mechanism allows to make use of the privacy enhancing techniques in an effective manner such that proportionate access of data is available to the observers whenever required. This chapter summarizes our research contributions and indicates some interesting directions in which our work can be extended.

## 6.1 Research Summary

In order to present an overall picture of security and privacy issues in video surveillance, we have presented an abstract model to systematically identify the security and privacy requirements in a video surveillance system. The two-stage process in our model enabled to identify a comprehensive set of requirements in all components of video surveillance, considering the perspective of each stakeholder involved. Examination of existing solutions, in multimedia systems and other relevant domains, against the identified requirements showed that these solutions cannot be readily used in video surveillance and hence further research efforts are required to devise security solutions in video surveillance. In this context, we have outlined a number of future research challenges, regarding confidentiality, integrity and access control, to be addressed for ensuring security and privacy in video surveillance systems.

From the security challenges that we identified, we have worked on the challenge of providing sufficient need-specific access of data to the observers yet protecting the privacy of people. We have proposed an access control model that integrates RBAC and ABAC bringing together the advantages offered by both models while addressing their limitations such as role-explosion. Though the model was developed considering the requirements of video surveillance environments, we first built a general-purpose model so that it may also be used in other applications sharing similar requirements. In the proposed model, the attributes may be associated with users, objects and environment thus allowing the request context to be considered when making access control decisions. Unlike traditional RBAC approaches, permissions in our model consist of operations and object expressions enabling content-based access control. We presented different request evaluation mechanisms that may be used by various applications, depending on their requirements, and presented two different algorithms to evaluate attribute-based requests using the proposed model. Our approach allows the evaluation of attribute-based requests by merely consulting the access control policy while retrieving the objects later only if the request is granted. This is an important characteristic of the proposed AERBAC model as a request may be denied without the overhead of retrieving the objects and the attributes associated with the user and the environment. We also demonstrated the merits of our model by comparing it with RBAC and ABAC using example configurations and discussing each model against the features desired in access control for video surveillance.

The proposed approach provides a comprehensive access control model offering following features: (1) the model allows to specify the access control policies based on the contents of the data to be protected. These content-dependent authorizations enable denoting a number of objects using a single permission

and permits to control access to objects which have not yet been created, based on the attributes that they may possess at the time of creation. This is particularly helpful in applications which contain a large number of objects and where the objects are added on a continuous basis, such as video surveillance systems; (2) in order to fully incorporate contextual information, the proposed approach allows to make context-aware decisions by associating attributes with all three entities: users, objects and environment; (3) the conditions linked to the permissions allow to specify constrained-permissions by using attributes of users, objects and environment. In certain applications, it is needed to compare values of object attributes and user attributes. Our approach allows to perform direct comparison between the attributes of the users and the objects which may significantly reduce the number of authorizations to be specified.

In order to meet the challenging requirements regarding specification and administration of data protection policies in video surveillance systems, we presented a multilevel access control mechanism for video surveillance environments based on our AERBAC model. In the process of creating resource object expressions, the video metadata, including the contents of the video, is also considered which allows representing a number of resource objects using a single permission. The presented solution is unique in terms of being role-oriented yet allowing specification of metadata-based access control policy. The privileges devised in our access control model are abstract and meaningful to interact with video surveillance data. As video data contains multiple levels of information, the privileges using our model may range from simply viewing the annotations associated with a video to accessing full information in a video. The concept of using privilege modes as a combination of video properties and actions enables multilevel access control without using negative authorizations. Besides role hierarchies, the existence of hierarchical relations among the attributes and partial order in the privilege modes allows derivation of further permissions from the explicitly specified permissions; this enables a larger set of authorizations comprised of relatively few authorizations. A prototype was implemented to show that the access control policies in video surveillance system using our mechanism can be specified via XACML, a standard policy specification language. Finally, a comparison of our approach with other relevant approaches showed that our mechanism is better-suited for video surveillance systems with respect to the features required in access control for video surveillance.

## 6.2 Thesis Contributions

- The introduction of an abstract model that is useful for finding a comprehensive set of security and privacy requirements in video surveillance systems.

- The identification of several research challenges regarding confidentiality, integrity and access control in video surveillance which are to be addressed when designing security solutions for video surveillance systems.

- Introduction of an access control model that combines the features offered by both RBAC and ABAC models and allows content-dependent policy specification yet providing administrative benefits such as modification visualization typically associated with RBAC.

- Evaluation of the AERBAC model against RBAC and ABAC models showing that the role-explosion and permission-explosion issues are resolved in AERBAC while allowing to take the contextual information into account.

- Development of a role-oriented multilevel access control mechanism for video surveillance environments making use of the metadata information associated with the resources.

- Implementation of the proposed access control mechanism using XACML to demonstrate specification and evaluation of the access control policies in video surveillance.

## 6.3 Future Research Directions

In this section, we provide a number of directions for extending the work presented in this thesis.

### 6.3.1 Administrative model

An interesting direction will be to develop an administrative model for our AERBAC model, similar to *ARBAC97* [123] defined for the RBAC model. The administrative model would allow to specify policies for the administrative personnel who may assign and revoke the roles as well as other attributes associated to the users, objects and environment. Some non-administrative users assuming certain roles may also be allowed to activate few of the environment attributes, the administrative model may also look into this aspect and allow specifying policies for such actions.

### 6.3.2 Formal Evaluation of AERBAC

In this thesis, we performed an initial evaluation of AERBAC against the RBAC and ABAC models showing that AERBAC model provides certain benefits not offered by either RBAC or ABAC while addressing the well-known problems of role-explosion and permission-explosion in RBAC. However, a formal evaluation of AERBAC against RBAC and ABAC requires significant research effort and remains to be done. The formal validation may work on quantifying how useful AERBAC is, compared to other models, in terms of expressiveness, granularity, scalability, etc. Moreover, performance of decision making may also be measured against other models for which a reference implementation of other models and appropriate policies would be needed.

### 6.3.3 Extend with Break-glass

The model that we propose allows the users to access resources in case the user holds certain attributes or if an environment attribute (e.g. occurrence of an event) is currently activated. However, it may not be possible to specify all the situations in the access control policy and there may be unknown critical situations not modeled in the policy when a user needs immediate access of relevant resources. Several techniques to incorporate break-glass in access control have been proposed [52, 30, 51]. Extending AERBAC with break-glass is an appealing research direction. Normally break-glass access allows a user to perform any action on any resources, though the user may be accountable for his/her actions later. AERBAC may offer an attractive benefit over other access control models in that it may restrict the break-glass access of a user to the resources represented by the permissions specified in user's roles rather than giving unrestricted access to all resources. In case of break-glass access, a user may be allowed to exercise those permissions existing in user's roles even if the conditions associated with the requested permissions are not fulfilled, since a break-glass access is typically needed in case of an abnormal situation which may not be represented by the specified conditions.

### 6.3.4 Continuous Enforcement

For continuous enforcement of access control, especially in case of access to live videos, further work is required to extend the proposed model with usage control mechanism [108] and enforcement mechanism for data streams [91, 90] such that the attributes which can change after granting access may be specified at the

policy level.

### 6.3.5 Provisioning and Disclosing of Attributes

Due to being a role-oriented approach, AERBAC can be used to specify constraints on provisioning of attributes to the users such that certain attributes may or may not be assigned to users when assuming certain roles. Similarly, in privacy-preserving authorization [11, 31, 154] where user needs to disclose possession of certain attributes to an authority in order to prove his/her access rights, constraints can be specified on roles stating which attributes can be disclosed when assuming a certain role.

Appendix A

# Legal Compliance

The legislation regarding video surveillance varies significantly in Europe as well as the rest of the world [111]. Some countries, e.g., Canada and Italy, have made regulations regarding usage of video surveillance by private and public authorities; others such as France and China have regulations but they apply mainly to private systems, while some countries, for example India, have no particular laws in this regard. Video surveillance is primarily criticized as a threat to privacy and hence it is mainly regulated in the context of privacy and data protection. We also suggest certain guidelines in order to help those who want to deploy video surveillance while least compromising the privacy of people and complying with legal infrastructure.

## A.1 Classification of Relevant Legislation

There could be several provisions related to video surveillance including privacy protection, criminal proceedings, federal & state laws and retention period which need to be consulted for employment of a video surveillance system in a particular country. We attempt to classify legislation regarding video surveillance into different categories, in order to serve as a reference point to investigate what relevant provisions are to be considered before deploying or for maintaining a

video surveillance systems in a particular region of the world.

#### A.1.0.1 Privacy and Data Protection

Most of the countries recognize the right to privacy. In some countries, for example the United States, where right of privacy is not recognized explicitly in the constitution there exist court rulings which recognize this right implicitly linking it with other provisions in the constitution. Regulations regarding privacy are particularly important in cases where specific regulations on video surveillance are missing in legislation of a country.

#### A.1.0.2 Communication Interception

Some countries such as Canada and Denmark have made regulations over communication interception, even by public or law enforcement authorities, and require a court order before intercepting the communication of an individual under surveillance. This court order is usually valid for a limited duration, few days to few months, and the court may renew it depending upon the matter, for example in criminal proceedings which could lead to prison of more than two years, for which an individual is being surveilled. This type of legislation is also particularly relevant if no or limited legislation exists regarding video surveillance.

#### A.1.0.3 Exemption for Public Authorities

In some cases where the system is to be operated by the public or law enforcement authorities, there might exist exemptions in the legislation. These exemptions may allow the authorities to perform communication interception or operate video surveillance systems to keep an eye over the activities of the general public or a specific target without any restrictions. For example, in France Police is allowed to remotely access and collect information held on IT systems.

#### A.1.0.4 Federal & State Laws

In countries such as the United States that consist of autonomous states/provinces, one needs to consult federal as well as state regulations in the context of video

surveillance. For example, few Canadian provinces have strict regulations and guidelines prepared by the provincial privacy commissioners that are to be followed by the private organizations who wish to deploy video surveillance systems in publicly accessible areas such as shopping malls and super markets.

#### A.1.0.5 Regulatory Body

Many countries have a regulatory body (also called regulator, privacy commissioner or data protection authority) to keep oversight of data protection practices being followed by the organizations that need to process personal data of any form whether images captured through video surveillance systems or other data such as medical and biometric data. In such countries, legislation might require registering and/or obtaining prior permission from regulator before employing video surveillance system operated by public or private authorities. In Spain, for example, the regulator requires an efficacy study of the system against alternative methods to justify a video surveillance system.

#### A.1.0.6 Video Surveillance

The legislation of a country may include laws that apply explicitly to video surveillance systems. The laws could be related to overt video surveillance systems or could cover covert systems as well which are normally used by security agencies or detectives in an investigation. Depending upon the video surveillance system to be deployed, one needs to consult the legislation carefully whether the laws apply to public authorities, private bodies or both. Countries such as Canada and the Netherlands which have explicit regulations regarding video surveillance often also include clauses about followings:

*Notification:* The requirement that the video surveillance system controller must notify the public about the surveillance by displaying meaningful symbols.

*Workplace Surveillance:* There might exist laws that refrain or restrict the usage of video surveillance systems over workplace to monitor employee performance.

*Retention Period:* The maximum time-limit for which the personal data or images could be stored.

*Privacy Safeguards:* The requirements regarding masking, logging, access control and auditing mechanism that limit access to the surveillance data.

*Public Access to their Data:* The law often requires that there must exist mechanism to allow people to access their images in a reasonable timeframe.

## A.2 Legislation in Selected Countries

In this section, we summarize the legislation of Canada, the United States and other selected countries in Europe [15] [111, 112] regarding privacy in video surveillance systems in the light of classification of relevant legislation, described in the previous section. We choose these countries because of two reasons. First, video surveillance is widely available in these countries. Secondly, there is sufficient information available regarding their regulations in English language, from reliable sources. This summary is not supposed to be exhaustive and the law of a country may include further requirements that are to be followed by the organizations who perform video surveillance. But it serves to provide a general idea.

#### A.2.0.7 Canada

| | |
|---|---|
| **1** | No explicit right of privacy in Charter of Rights and Freedoms, although it outlines protection from unreasonable search and seizure which is often considered to be applicable on informational privacy too |
| **2** | Communication interception requires court order |
| **3** | The federal Personal Information Protection and Electronic Documentation Act (PIPEDA) also applies to video surveillance |
| **4** | Regulatory body does exist and has provided certain guidelines both for covert and overt video surveillance performed by public and private sector |
| **5** | It is obligatory to inform public about video surveillance via signs |
| **6** | The signage should include the purpose of collection of video surveillance and the organization's privacy contact person |
| **7** | Video data should be kept only as long as necessary and must be destroyed when no more required |

### A.2.0.8 Denmark

| | |
|---|---|
| **1** | The right of privacy is recognized in the constitution |
| **2** | Communication interception requires court order and the frequency of requests and approval is quite high |
| **3** | It is obligatory to inform public about video surveillance via signs |
| **4** | In 2007, act on TV surveillance was amended enabling private sector to perform video surveillance on their property which was previously not allowed |
| **5** | There exists a regulatory body, however, informing the regulator or taking approval is not required before installing video surveillance systems |
| **6** | It is obligatory to inform public about video surveillance via signs |
| **7** | Retention period for video data is not clearly stated, however, retention period for other personal information is 12 months |

### A.2.0.9 France

| | |
|---|---|
| **1** | Privacy right is not explicitly mentioned in the constitution but has been ruled to be implicit |
| **2** | Police is allowed to remotely access, record, collect and transfer information held on IT systems |
| **3** | Anti-terror act of 2006 authorizes private parties to install video surveillance on places open to public and likely to be exposed to risks of aggression or theft |
| **4** | Regulatory body exists and authorization from regulatory body is required before installing video surveillance systems . However, it has limited powers over activities of government |
| **5** | It is obligatory to inform public about video surveillance via signs |
| **6** | Retention period for video data is not clearly stated, however, retention period for other personal information is 12 months |

### A.2.0.10 Italy

| | |
|---|---|
| **1** | No explicit protection of privacy in the constitution, though protections for communication and home are there |
| **2** | Pre-emptive communication interception may occur at the discretion of Attorney General |
| **3** | Strong regulatory authority exists |
| **4** | Video surveillance in public places is permitted only if it is proportionate to the pursuing objectives and should only be activated when other measures are inadequate |
| **5** | It is obligatory to inform public about video surveillance via signage |
| **6** | Storage of images should be limited in time |

### A.2.0.11 Netherlands

| | |
|---|---|
| **1** | The right of privacy is recognized in the constitution |
| **2** | Communication interception requires court order |
| **3** | Video surveillance in public places requires informing the regulator in advance |
| **4** | It is obligatory to inform public about video surveillance via signs |
| **5** | Video data could be retained for upto four weeks |

### A.2.0.12 Norway

| | |
|---|---|
| **1** | Constitution does not include a specific privacy clause |
| **2** | Communication interception requires judicial warrant, however, bugging conversations of criminals by police is relaxed |
| **3** | Regulator operates under ministry of Government Administration but is generally considered independent |
| **4** | There is no requirement to inform regulatory body in case of non-recorded video surveillance. However, for recorded video surveillance, the regulatory body has to be informed which has the power to prohibit video surveillance |
| **5** | It is obligatory to inform public about video surveillance via signs |
| **6** | No data retention law exists |

### A.2.0.13 Spain

| | |
|---|---|
| **1** | The right of privacy is recognized in the constitution |
| **2** | Communication interception does not required court order |
| **3** | Video surveillance can only be used when other proportionate methods are not available |
| **4** | Video surveillance has to be reported to regulator who will assess its justification |
| **5** | It is obligatory to inform public about video surveillance via signs |
| **6** | Video data has to be removed after one month |

### A.2.0.14 United Kingdom

| | |
|---|---|
| **1** | No constitutional right of privacy |
| **2** | Communication interception does not require court order rather ministerial approval is enough |
| **3** | Surveillance can be done by Police, local authorities or private sector |
| **4** | Regulator has been granted greater powers and fining capacities |
| **5** | It is obligatory to inform public about video surveillance via signs |
| **6** | Video data could be retained for two years |
| **7** | Although video surveillance code of practices exist but they have no legal binding |

### A.2.0.15 United States

| | |
|---|---|
| **1** | No right to privacy in constitution, though court has ruled linking it with other provisions |
| **2** | Data Privacy Act protects records held by public authorities, but no comprehensive data protection law for private sector exists |
| **3** | Federal Trade Commission issued self-regulating privacy guidelines, however, it has no authority to enforce privacy rights |
| **4** | No federal law regarding video surveillance exists. Video surveillance laws in different states vary, for example, in New York video surveillance can only be conducted by the police while in Arizona one can use video surveillance at a public place without posting a notice to inform public |
| **5** | No data retention law exists |

## A.3 Guidelines

Many public and private organizations have provided certain guidelines regarding deployment and maintenance of the video surveillance systems. Some focus only on public sector and law enforcement agencies, e.g, guidelines given by Beech et al. [140], while others target both public and private organizations which are using video surveillance systems, e.g., [49]. Whether these guidelines are legally binding or not depends on the fact whether they have been provided by a private organization or a public authority, for example, there exist certain guidelines in Canada regarding video surveillance which are provided by the Privacy Commissioner of Canada and hence are legally binding. Here we combine the guidelines for employing video surveillance, provided by European Data Protection Supervisor [49], Privacy Commissioner of Canada [110], and Constitution Project [140] – a non-government organization in USA. These recommendations, albeit not guarantee to avoid breach of law, ensure that the impact of video surveillance on privacy is minimized and therefore may help achieving the compliance with the privacy legislation in a particular country.

1. Establish a lawful reason for conducting video surveillance and use video surveillance only for that purpose
2. Determine whether a less privacy-invasive alternative to video surveillance would meet the requirements
3. Perform the cost-benefit analysis, comparing the alternative means of addressing the stated purpose of the system
4. Build privacy into the system design and address data protection issues on early stage
5. Assess the impact of system on privacy and freedom of individuals
6. Consult the regulatory authority – if any – and other stakeholders of the system, for example employee representative in case of workplace monitoring
7. Determine whether live monitoring without recording is enough, otherwise, store the recorded images securely and destroy them after a specified time (one week in most of the cases)
8. Provide public notices about the surveillance
9. Devise a mechanism to give individuals access to data about them
10. Create technological and administrative safeguards to reduce the possibility of misuse and abuse of the system

11. Provide training to the system operators and educate them on obligation to protect the privacy of individuals
12. Maintain a secure log in order to keep track of the activities performed by the system operators

# Bibliography


[1] Directive 2006/24/ec of the european parliament and of the council of 15 march 2006. `http://www.enisa.europa.eu/activities/risk-management/current-risk/laws-regulation/national-security/directive-2006-24-ec`, 2006. Last accessed: December, 2015.

[2] Xacml light. `http://xacmllight.sourceforge.net/`, 2011. Last accessed: December, 2015.

[3] Xacml enterprise. `https://code.google.com/p/enterprise-java-xacml/`, 2012. Last accessed: December, 2015.

[4] Adam, N. R., Atluri, V., Bertino, E., and Ferrari, E. A content-based authorization model for digital libraries. *IEEE Transactions on Knowledge and Data Engineering 14*, 2 (2002), 296–315.

[5] Aich, S., Sural, S., and Majumdar, A. K. Starbac: Spatiotemporal role based access control. In *On the Move to Meaningful Internet Systems* (2007), Springer, pp. 1567–1582.

[6] Al-Kahtani, M., Sandhu, R., et al. A model for attribute-based user-role assignment. In *Annual Computer Security Applications Conference* (2002), IEEE, pp. 353–362.

[7] Albrechtslund, A. Online social networking as participatory surveillance. *First Monday 13*, 3 (2008).

[8] Alshehri, S., and Raj, R. K. Secure access control for health information sharing systems. In *International Conference on Healthcare Informatics* (2013), IEEE, pp. 277–286.

[9] American-Management-Association. Electronic monitoring & surveillance survey. http://www.plattgroupllc.com/jun08/2007ElectronicMonitoringSurveillanceSurvey.pdf, 2007. Last accessed: December, 2015.

[10] Applied-Autonomy-Institute. i-see 'now more than ever'. http://www.appliedautonomy.com/isee.html, 2005. Last accessed: December, 2015.

[11] Ardagna, C. A., De Capitani di Vimercati, S., Neven, G., Paraboschi, S., Preiss, F.-S., Samarati, P., and Verdicchio, M. Enabling privacy-preserving credential-based access control with xacml and saml. In *International Conference on Computer and Information Technology (CIT)* (2010), IEEE, pp. 1090–1095.

[12] Atluri, V., and Chun, S. An authorization model for geospatial data. *IEEE Transactions on Dependable and Secure Computing 1*, 4 (2004), 238–254.

[13] Atluri, V., and Chun, S. A. A geotemporal role-based authorisation system. *International Journal of Information and Computer Security 1*, 2 (2007), 143–168.

[14] Atrey, P. K., Yan, W.-Q., and Kankanhalli, M. S. A scalable signature scheme for video authentication. *Journal of Multimedia Tools and Applications 34*, 1 (2007), 107–135.

[15] Banisar, D., and Davies, S. G. Global trends in privacy protection: An international survey of privacy, data protection, and surveillance laws and developments. *John Marshall Journal of Computer & Information Law 18*, 1 (1999), 3–111.

[16] Basharat, A., Gritai, A., and Shah, M. Learning object motion patterns for anomaly detection and improved object detection. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (2008), IEEE, pp. 1–8.

[17] Başтan, M., Cam, H., Güdükbay, U., and Ulusoy, Ö. An mpeg-7 compatible video retrieval system with integrated support for complex multimodal queries. *IEEE Multimedia 17*, 3 (2009), 1–30.

[18] BBC-News. Cctv: Does it work? http://news.bbc.co.uk/1/hi/uk/2071496.stm, 2002. Last accessed: December, 2015.

[19] BBC-News. Cctv staff 'spied on naked woman. http://news.bbc.co.uk/2/hi/uk_news/england/merseyside/4503244.stm, 2005. Last accessed: December, 2015.

[20] BBC-News. Edward snowden: Timeline. `http://www.bbc.com/news/world-us-canada-23768248`, 2013. Last accessed: December, 2015.

[21] Bertino, E., Bonatti, P. A., and Ferrari, E. Trbac: A temporal role-based access control model. *ACM Transactions on Information and System Security (TISSEC) 4*, 3 (2001), 191–233.

[22] Bertino, E., Catania, B., Damiani, M. L., and Perlasca, P. Geo-rbac: a spatially aware rbac. In *Symposium on Access Control Models and technologies* (2005), ACM, pp. 29–37.

[23] Bertino, E., Fan, J., Ferrari, E., Hacid, M.-S., Elmagarmid, A. K., and Zhu, X. A hierarchical access control model for video database systems. *ACM Transactions on Information Systems (TOIS) 21*, 2 (2003), 155–191.

[24] Bertino, E., Hammad, M. A., Aref, W. G., and Elmagarmid, A. K. An access control model for video database systems. In *9th International Conference on Information and Knowledge Management* (2000), ACM, pp. 336–343.

[25] Birnstill, P., and Pretschner, A. Enforcing privacy through usage-controlled video surveillance. In *International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2013), IEEE, pp. 318–323.

[26] Blaze, M., Feigenbaum, J., and Lacy, J. Decentralized trust management. In *Symposium on Security and Privacy* (1996), IEEE, pp. 164–173.

[27] Boult, T. E. Pico: Privacy through invertible cryptographic obscuration. In *Computer Vision for Interactive and Intelligent Environment* (2005), IEEE, pp. 27–38.

[28] Brin, D. *The Transparent Society: Will Technology Force Us to Choose Between Privacy and Freedom.* Perseus Publishing, 1999.

[29] Bringel Filho, J., and Martin, H. A generalized context-based access control model for pervasive environments. In *International Workshop on Security and Privacy in GIS and LBS* (2009), ACM, pp. 12–21.

[30] Brucker, A. D., and Petritsch, H. Extending access control models with break-glass. In *Symposium on access control models and technologies* (2009), ACM, pp. 197–206.

[31] Camenisch, J., Mödersheim, S., Neven, G., Preiss, F.-S., and Sommer, D. A card requirements language enabling privacy-preserving access control. In *Symposium on access control models and technologies* (2010), ACM, pp. 119–128.

[32] Carrillo, P., Kalva, H., and Magliveras, S. Compression independent object encryption for ensuring privacy in video surveillance. In *International Conference on Multimedia and Expo* (2008), IEEE, pp. 273–276.

[33] Castiglione, A., Cepparulo, M., De Santis, A., and Palmieri, F. Towards a lawfully secure and privacy preserving video surveillance system. In *11th International Conference on E-Commerce and Web Technologies* (2010), Springer, pp. 73–84.

[34] Cavallaro, A. Privacy in video surveillance. *IEEE Signal Processing Magazine 24*, 2 (2007), 168–169.

[35] Chae, J. H., and Shiri, N. Formalization of rbac policy with object class hierarchy. In *Information Security Practice and Experience* (2007), Springer, pp. 162–176.

[36] Covington, M. J., Long, W., Srinivasan, S., Dev, A. K., Ahamad, M., and Abowd, G. D. Securing context-aware applications using environment roles. In *Symposium on Access Control Models and Technologies* (2001), ACM, pp. 10–20.

[37] Covington, M. J., and Sastry, M. R. A contextual attribute-based access control model. In *On the Move to Meaningful Internet Systems* (2006), Springer, pp. 1996–2006.

[38] Coyne, E., and Weil, T. R. Abac and rbac: Scalable, flexible, and auditable access management. *IT Professional 15*, 3 (2013), 14–16.

[39] Coyne, E. J., Weil, T. R., and Kuhn, R. Role engineering: Methods and standards. *IT Professional 13*, 6 (2011), 54–57.

[40] Criminisi, A., Perez, P., and Toyama, K. Object removal by exemplar-based inpainting. In *13th IEEE Computer Visioin and Pattern Recognition* (2003), IEEE, pp. 721–728.

[41] Criminisi, A., Pérez, P., and Toyama, K. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing 13*, 9 (2004), 1200–1212.

[42] Damiani, E., di Vimercati, S. D. C., and Samarati, P. New paradigms for access control in open environments. In *International Symposium on Signal Processing and Information Technology* (2005), IEEE, pp. 540–545.

[43] David, F., and Richard, K. Role-based access controls. In *15th NIST-NCSC National Computer Security Conference* (1992), NIST, pp. 554–563.

[44] Döller, M., and Kosch, H. The mpeg-7 multimedia database system (mpeg-7 mmdb). *Journal of Systems and Software 81*, 9 (2008), 1559–1580.

[45] Draper, T. An introduction to jeremy bentham's theory of punishment. *ournal of Bentham Studies 5*, 1 (2002), 1–17.

[46] Dufaux, F., and Ebrahimi, T. Scrambling for privacy protection in video surveillance systems. *IEEE Transactions on Circuits and Systems for Video Technology 18*, 8 (2008), 1168–1174.

[47] Emerson, T. I. *The system of freedom of expression.* Random House Trade, 1970.

[48] EmpowerID. Best practices in enterprise authorization: The rbac/abac hybrid approach. `http://blog.empowerid.com/Portals/174819/docs/EmpowerID-WhitePaper-RBAC-ABAC-Hybrid-Model.pdf`, 2013. Last accessed: December, 2015.

[49] European-Data-Protection-Supervisor. The edps video surveillance guidelines. `https://secure.edps.europa.eu/EDPSWEB/webdav/site/mySite/shared/Documents/Supervision/Guidelines/10-03-17_Video-surveillance_Guidelines_EN.pdf`, 2010. Last accessed: December, 2015.

[50] Ferraiolo, D. F., Sandhu, R., Gavrila, S., Kuhn, D. R., and Chandramouli, R. Proposed nist standard for role-based access control. *ACM Transactions on Information and System Security (TISSEC) 4*, 3 (2001), 224–274.

[51] Ferreira, A., Chadwick, D., Farinha, P., Correia, R., Zao, G., Chilro, R., and Antunes, L. How to securely break into rbac: the btg-rbac model. In *Annual Computer Security Applications Conference (ACSAC)* (2009), IEEE, pp. 23–31.

[52] Ferreira, A., Cruz-Correia, R., Antunes, L. H. M., Farinha, P., Oliveira-Palhares, E., Chadwick, D. W., and Costa-Pereira, A. How to break access control in a controlled manner. In *International Symposium on Computer-Based Medical Systems (CBMS)* (2006), IEEE, pp. 847–854.

[53] Foucault, M. *Discipline and punishment.* New York: Pantheon, 1977.

[54] Franqueira, V. N. L., and Wieringa, R. J. Role-based access control in retrospect. *IEEE Computer 45*, 6 (2012), 81–88.

[55] Ge, M., and Osborn, S. L. A design for parameterized roles. In *International Data, Application Security and Privacy Conference* (2004), Springer, pp. 251–264.

[56] Georgiadis, C. K., Mavridis, I., Pangalos, G., and Thomas, R. K. Flexible team-based access control using contexts. In *Proceedings of the 6th ACM symposium on Access control Models and Technologies* (2001), ACM, pp. 21–27.

[57] Giuri, L., and Iglio, P. Role templates for content-based access control. In *Workshop on Role-Based Access Control* (1997), ACM, pp. 153–159.

[58] Hampapur, A. Smart video surveillance for proactive security. *IEEE Signal Processing Magazine 25*, 4 (2008), 131–134.

[59] Hampapur, A., Brown, L., Feris, R., Senior, A., Shu, C.-F., Tian, Y., Zhai, Y., and Lu, M. Searching surveillance video. In *International Conference on Advanced Video and Signal Based Surveillance* (2007), IEEE, pp. 75–80.

[60] Hansen, F., and Oleshchuk, V. Srbac: A spatial role-based access control model for mobile systems. In *Proceedings of the 7th Nordic Workshop on Secure IT Systems (NORDSEC'03)* (2003), Citeseer, pp. 129–141.

[61] Haritaoglu, I., Harwood, D., and Davis, L. S. W4: Real-time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence 22*, 8 (2000), 809–830.

[62] Haskell, B. G., Puri, A., and Netravali, A. N. *Digital Video: an Introduction to MPEG-2*. Springer Science & Business Media, 1997.

[63] Huang, J., Nicol, D. M., Bobba, R., and Huh, J. H. A framework integrating attribute-based policies into role-based access control. In *Symposium on Access Control Models and Technologies* (2012), ACM, pp. 187–196.

[64] Ingwar, M., Ahmed, N., and Jensen, C. D. Error-rate-based fusion of biometric experts. In *International Conference on Privacy, Security and Trust (PST)* (2013), IEEE, pp. 239–246.

[65] Jensen, C. D., Geneser, K., and Willemoes-Wissing, I. C. Sensor enhanced access control: extending traditional access control models with context-awareness. In *7th IFIP International Conference on Trust Management (IFIPTM)* (2013), Springer, pp. 177–192.

[66] Jin, X., Krishnan, R., and Sandhu, R. S. A unified attribute-based access control model covering dac, mac and rbac. In *26th Data and Applications Security and Privacy Conference* (2012), Springer, pp. 41–55.

[67] Jin, X., Sandhu, R., and Krishnan, R. Rabac: role-centric attribute-based access control. In *Mathematical Methods, Models and Architectures for Computer Network Security* (2012), Springer, pp. 84–96.

[68] Joshi, J. B., Bertino, E., Latif, U., and Ghafoor, A. A generalized temporal role-based access control model. *IEEE Transactions on Knowledge and Data Engineering 17*, 1 (2005), 4–23.

[69] Kalam, A. A. E., Baida, R., Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., Miege, A., Saurel, C., and Trouessin, G. Organization based access control. In *International Workshop on Policies for Distributed Systems and Networks* (2003), IEEE, pp. 120–131.

[70] Kern, A., and Walhorn, C. Rule support for role-based access control. In *Symposium on Access Control Models and Technologies* (2005), ACM, pp. 130–138.

[71] Kim, Y.-G., Mon, C.-J., Jeong, D., Lee, J.-O., Song, C.-Y., and Baik, D.-K. *Context-aware access control mechanism for ubiquitous applications.* Advances in Web Intelligence. Springer, 2005, pp. 236–242.

[72] Kodali, N., Farkas, C., and Wijesekera, D. An authorization model for multimedia digital libraries. *International Journal on Digital Libraries 4*, 3 (2004), 139–155.

[73] Koshimizu, T., Toriyama, T., and Babaguchi, N. Factors on the sense of privacy in video surveillance. In *Workshop on Continuous Archival and Retrival of Personal Experiences* (2006), ACM, pp. 35–44.

[74] Kuhn, D. R., Coyne, E. J., and Weil, T. R. Adding attributes to role-based access control. *IEEE Computer 43*, 6 (2010), 79–81.

[75] Kulkarni, D., and Tripathi, A. Context-aware role-based access control in pervasive computing systems. In *Symposium on Access Control Models and Technologies* (2008), ACM, pp. 113–122.

[76] Kumagai, J., and Cherry, S. Sensors & sensibility costs, convenience, and security all converge on this. *IEEE SPECTRUM 41*, 7 (2004), 18–24.

[77] Kumar, M., and Newman, R. E. Strbac-an approach towards spatio-temporal role-based access control. In *Communication, Network, and Information Security* (2006), pp. 150–155.

[78] Lazouski, A., Martinelli, F., and Mori, P. Usage control in computer security: A survey. *Computer Science Review 4*, 2 (2010), 81–99.

[79] Lin, C.-Y., Tseng, B. L., and Smith, J. R. Videoannex: Ibm mpeg-7 annotation tool for multimedia indexing and concept learning. In *IEEE International Conference on Multimedia and Expo* (2003), pp. 1–2.

[80] Liu, F., and Koenig, H. Puzzle–a novel video encryption algorithm. In *9th International Conference on Communications and Multimedia Security* (2005), Springer, pp. 88–97.

[81] Liu, F., and Koenig, H. A survey of video encryption algorithms. *Journal of computers & security 29*, 1 (2010), 3–15.

[82] Liu, X., and Eskicioglu, A. M. Selective encryption of multimedia content in distribution networks: Challenges and new directions. In *2nd IASTED Conference on Communications, Internet & Information Technology* (2003), pp. 527–533.

[83] Liu, Z., Peng, D., Zheng, Y., and Liu, J. Communication protection in ip-based video surveillance systems. In *7th IEEE International Symposium on Multimedia* (2005), IEEE, pp. 69–78.

[84] Lyon, D. Biometrics, identification and surveillance. *Bioethics 22*, 9 (2008), 499–508.

[85] Maguire, M. Restraining big brother? the regulation of surveillance in england and wales. In *Surveillance, Closed Circuit Television and Social Control* (1998), Ashgate Publishing Ltd., pp. 229–240.

[86] Marx, G. T., and Muschert, G. W. Personal information, borders, and the new surveillance studies. *Annual Review of Law and Social Science 3*, 1 (2007), 375–395.

[87] Moncrieff, S., Venkatesh, S., and West, G. A. Dynamic privacy in public surveillance. *Journal of Computer 42*, 4 (2009), 22–28.

[88] Moyer, M. J., and Ahamad, M. Generalized role-based access control. In *21st International Conference on Distributed Computing Systems* (2001), IEEE, pp. 391–398.

[89] Nagel, T. *Concealment and exposure: and other essays*. Oxford University Press, USA, 2002.

[90] Nehme, R. V., Lim, H.-S., and Bertino, E. Fence: Continuous access control enforcement in dynamic data stream environments. In *Conference on data and application security and privacy* (2013), ACM, pp. 243–254.

[91] Nehme, R. V., Rundensteiner, E. A., and Bertino, E. A security punctuation framework for enforcing access control on streaming data. In *International Conference on Data Engineering (ICDE)* (2008), IEEE, pp. 406–415.

[92] Neumann, G., and Strembeck, M. An approach to engineer and enforce context constraints in an rbac environment. In *Symposium on access control models and technologies* (2003), ACM, pp. 65–79.

[93] New-York-Times. Chicago moving to 'smart' surveillance cameras. `http://www.nytimes.com/2004/09/21/national/21cameras.html`, 2004. Last accessed: December, 2015.

[94] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Symposium on Security and Privacy* (2013), IEEE, pp. 541–555.

[95] Norris, C., and Armstrong, G. Cctv and the social structuring of surveillance. In *Surveillance of public space: CCTV, street lighting and crime prevention* (1999), Criminal Justice Press, pp. 157–178.

[96] Norris, C., and Armstrong, G. *The Maximum Surveillance Society.* Berg, 1999.

[97] Norris, C., McCahill, M., and Wood, D. The growth of cctv: a global perspective on the international diffusion of video surveillance in publicly accessible space. *Surveillance & Society 2*, 2/3 (2004), 110–135.

[98] OASIS. Security assertion markup language, version 2.0. `http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf`, 2005. Last accessed: December, 2015.

[99] OASIS. Web services federation language (ws-federation), version 1.2. `http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html`, 2009. Last accessed: December, 2015.

[100] OASIS. Xacml profile for role based access control (rbac). `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-rbac-v1-spec-en.pdf`, 2014. Last accessed: December, 2015.

[101] OASIS-Standard. Security assertion markup language. `http://saml.xml.org/wiki/saml-introduction`, 2005. Last accessed: March, 2013.

[102] OASIS-Standard. Extensible access control markup language (xacml) version 3.0. `http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf`, 2013. Last accessed: December, 2015.

[103] Observing-Surveillance. Observing surveillance project in washington. `http://observingsurveillance.org/`, 2002. Last accessed: December, 2015.

[104] O'Connor, A. C., and Loomis, R. J. Economic analysis of role-based access control, nist report. `http://csrc.nist.gov/groups/SNS/rbac/documents/20101219_RBAC2_Final_Report.pdf`, 2010. Last accessed: December, 2015.

[105] O'Donnell, A. T. *Who is watching you, and why? A social identity analysis of surveillance.* PhD thesis, University of Exeter, UK, 2010.

[106] Osborn, S., Sandhu, R., and Munawer, Q. Configuring role-based access control to enforce mandatory and discretionary access control policies. *ACM Transactions on Information and System Security (TISSEC) 3*, 2 (2000), 85–106.

[107] Pan, L., and Zhang, C. N. A web-based multilayer access control model for multimedia applications in mpeg-7. *International Journal of Network Security 4*, 2 (2007), 155–165.

[108] Park, J., and Sandhu, R. The ucon abc usage control model. *ACM Transactions on Information and System Security (TISSEC) 7*, 1 (2004), 128–174.

[109] Presti, L. L., and Cascia, M. L. Real-time object detection in embedded video surveillance systems. In *Image Analysis for Multimedia Interactive Services* (2008), IEEE, pp. 151–154.

[110] Privacy-Commissioner-Canada. Guidelines for overt video surveillance in private sector. `https://www.priv.gc.ca/information/guide/2008/gl_vs_080306_e.asp`, 2008. Last accessed: December, 2015.

[111] Privacy-International. Surveillance monitor 2007 – international country rankings. `https://www.privacyinternational.org/reports/surveillance-monitor-2007-international-country-rankings`, 2007. Last accessed: December, 2015.

[112] Privacy-International. European privacy and human rights. `https://www.privacyinternational.org/sites/privacyinternational.org/files/file-downloads/ephr.pdf`, 2010. Last accessed: December, 2015.

[113] Rajpoot, Q. M., and Jensen, C. D. Security and privacy in video surveillance: Requirements and challenges. In *29th IFIP International Information Security and Privacy Conference (SEC)* (2014), Springer, pp. 169–184.

[114] Rajpoot, Q. M., and Jensen, C. D. Video surveillance: Privacy issues and legal compliance. In *Promoting Social Change and Democracy Through Information Technology*, V. Kumar and J. Svensson, Eds. IGI Global, 2015, pp. 69–92.

[115] Rajpoot, Q. M., and Jensen, C. D. Role-oriented access control model for video surveillance. *Elsevier Computers and Security* (To be submitted).

[116] Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Attributes enhanced role-based access control model. In *12th International Conference*

*on Trust, Privacy and Security in Digital Business (TrustBus)* (2015), Springer, pp. 3–17.

[117] Rajpoot, Q. M., Jensen, C. D., and Krishnan, R. Integrating attributes into role-based access control. In *29th Data and Applications Security and Privacy Conference (DBSec)* (2015), Springer, pp. 242–249.

[118] Ray, I., Kumar, M., and Yu, L. *LRBAC: a location-aware role-based access control model.* International Conference on Information Systems Security (ICISS). Springer, 2006, pp. 147–161.

[119] Ray, I., and Toahchoodee, M. A spatio-temporal role-based access control model. In *Data and Applications Security and Privacy Conference* (2007), Springer, pp. 211–226.

[120] Ray, I., and Toahchoodee, M. A spatio-temporal role-based access control model. In *International Conference on Data and Applications Security.* Springer, 2007, pp. 211–226.

[121] Rosen, J. *The Naked Crowd: Reclaiming Security and Freedom in an Anxious Age.* Random House Incorporated, 2005.

[122] Saini, M. K., Atrey, P. K., Mehrotra, S., and Kankanhalli, M. S. Privacy aware publication of surveillance video. *International Journal of Trust Management in Computing and Communications 1*, 1 (2013), 23–51.

[123] Sandhu, R., Bhamidipati, V., and Munawer, Q. The arbac97 model for role-based administration of roles. *ACM Transactions on Information and System Security (TISSEC) 2*, 1 (1999), 105–135.

[124] Sandhu, R. S., Coyne, E. J., Feinstein, H. L., and Youman, C. E. Role-based access control models. *Computer*, 2 (1996), 38–47.

[125] Schaad, A., Moffett, J., and Jacob, J. The role-based access control system of a european bank: a case study and discussion. In *Symposium on Access Control Models and Technologies* (2001), ACM, pp. 3–9.

[126] Schaffer, M., and Schartner, P. Video surveillance: a distributed approach to protect privacy. In *9th International Conference on Communications and Multimedia Security* (2005), Springer, pp. 140–149.

[127] Schiff, J., Meingast, M., Mulligan, D. K., Sastry, S., and Goldberg, K. Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Protecting Privacy in Video Surveillance* (2009), Springer, pp. 65–89.

[128] Schneider, M., and Chang, S.-F. A robust content based digital signature for image authentication. In *International Conference on Image Processing* (1996), IEEE, pp. 227–230.

[129] Semple, J. *Bentham's Prison: A Study of the Panopticon Penitentiary.* Oxford University Press, 1993.

[130] Senior, A., Pankanti, S., Hampapur, A., Brown, L., Tian, Y.-L., Ekin, A., Connell, J., Shu, C. F., and Lu, M. Enabling video privacy through computer vision. *Journal of Security & Privacy 3*, 3 (2005), 50–57.

[131] Shen, H.-b., and Hong, F. An attribute-based access control model for web services. In *Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT'06. Seventh International Conference on* (2006), IEEE, pp. 74–79.

[132] Slobogin, C. Public privacy: camera surveillance of public places and the right to anonymity. *Mississippi Law Journal 72*, 1 (2002), 213–233.

[133] Socek, D., Magliveras, S., Ćulibrk, D., Marques, O., Kalva, H., and Furht, B. Digital video encryption algorithms based on correlation-preserving permutations. *EURASIP Journal on Information Security 2007*, 10 (2007), 1–15.

[134] Solove, D. J. Digital dossiers and the dissipation of fourth amendment privacy. *Southern California Law Review 75*, 1 (2002), 1083–1169.

[135] Solove, D. J. I have got nothing to hide, and other misunderstandings of privacy. *San Diego law review 44*, 1 (2007), 745–772.

[136] Solove, D. J. *Nothing to hide: The false tradeoff between privacy and security.* Yale University Press, 2011.

[137] Sun, Q., He, D., and Tian, Q. A secure and robust authentication scheme for video transcoding. *IEEE Transactions on Circuits and Systems for Video Technology 16*, 10 (2006), 1232–1244.

[138] Systems, S. Sun's xacml implementation. `http://sunxacml.sourceforge.net/`, 2006. Last accessed: December, 2015.

[139] Tang, F., Ying, Y., Wang, J., and Peng, Q. A novel texture synthesis based algorithm for object removal in photographs. In *9th Asian Computing Science Conference* (2004), Springer, pp. 248–258.

[140] The-Constitution-Project. Guidelines for public video surveillance: A guide to protecting communities and preserving civil liberties. `http://www.constitutionproject.org/pdf/Video_Surveillance_`

Guidelines_Report_w_Model_Legislation4.pdf, 2006. Last accessed: December, 2015.

[141] The-Guardian. Airport worker given police warning for 'misusing' body scanner. http://www.theguardian.com/uk/2010/mar/24/airport-worker-warned-body-scanner, 2010. Last accessed: December, 2015.

[142] The-Guardian. Edward snowden: the whistleblower behind the nsa surveillance revelations. http://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance, 2013. Last accessed: December, 2015.

[143] The-US-Small-Business-Administration. Email, phone and social media monitoring in the workplace – know your rights as an employer. https://www.sba.gov/blogs/email-phone-and-social-media-monitoring-workplace-know-your-rights-employer 2012. Last accessed: December, 2015.

[144] Thornton, J., Baran-Gale, J., Butler, D., Chan, M., and Zwahlen, H. Person attribute search for large-area video surveillance. In *International Conference on Technologies for Homeland Security* (2011), IEEE, pp. 55–61.

[145] Thuraisingham, B., Lavee, G., Bertino, E., Fan, J., and Khan, L. Access control, confidentiality and privacy for video surveillance databases. In *11th ACM Symposium on Access Control Models and Technologies* (2006), ACM, pp. 1–10.

[146] Tian, Y., Feris, R. S., Liu, H., Hampapur, A., and Sun, M.-T. Robust detection of abandoned and removed objects in complex surveillance videos. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, 41*, 5 (2011), 565–576.

[147] Ullmann-Margalit, E. The case of the camera in the kitchen: Surveillance, privacy, sanctions, and governance. *Regulation & Governance 2*, 4 (2008), 425–444.

[148] Vaquero, D., Feris, R. S., Tran, D., Brown, L., Hampapur, A., Turk, M., et al. Attribute-based people search in surveillance environments. In *Applications of Computer Vision* (2009), IEEE, pp. 1–8.

[149] Westermann, U., and Klas, W. Ptdom: a schema-aware xml database system for mpeg-7 media descriptions. *Software: Practice and Experience 36*, 8 (2006), 785–834.

[150] Westin, A. F. *Privacy and freedom.* New York: Athenum, 1970.

[151] Winkler, T., and Rinner, B. A systematic approach towards user-centric privacy and security for smart camera networks. In *4th ACM/IEEE International Conference on Distributed Smart Cameras* (2010), ACM, pp. 133–141.

[152] Winkler, T., and Rinner, B. Trustcam: Security and privacy-protection for an embedded smart camera based on trusted computing. In *7th IEEE Advanced Video and Signal Based Surveillance* (2010), IEEE, pp. 593–600.

[153] Winkler, T., and Rinner, B. Securing embedded smart cameras with trusted computing. *EURASIP Journal on Wireless Communications and Networking 2011*, 8 (2011), 1–20.

[154] Winsborough, W. H., and Li, N. Towards practical automated trust negotiation. In *International Workshop on Policies for Distributed Systems and Networks* (2002), IEEE, pp. 92–103.

[155] Winslett, M., Ching, N., Jones, V. E., and Slepchin, I. Using digital credentials on the world wide web. *Journal of Computer Security 5*, 3 (1997), 255–266.

[156] Wood, D. Foucault and panopticism revisited. *Surveillance & Society 1*, 3 (2002), 234–239.

[157] WSO2. Xacml 3.0 implementation – balana. `http://xacmlinfo.org/2012/08/16/balana-the-open-source-xacml-3-0-implementation/`, 2013. Last accessed: December, 2015.

[158] Xu, Z., and Stoller, S. D. Mining attribute-based access control policies from rbac policies. In *10th International Conference and Expo on Emerging Technologies for a Smarter World (CEWIT)* (2013), IEEE, pp. 1–6.

[159] Yen, T.-F., Xie, Y., Yu, F., Yu, R. P., and Abadi, M. Host fingerprinting and tracking on the web: Privacy and security implications. In *Network and Distributed System Security Symposium* (2012), Internet Society, pp. 1–16.

[160] Yu, X., Chinomi, K., Koshimizu, T., Nitta, N., Ito, Y., and Babaguchi, N. Privacy protecting visual processing for secure video surveillance. In *International Conference on Image Processing* (2008), IEEE, pp. 1672–1675.

[161] Yuan, E., and Tong, J. Attributed based access control (abac) for web services. In *International Conference on Web Services* (2005), IEEE, pp. 561–569.

[162] Zhang, G., and Parashar, M. Context-aware dynamic access control for pervasive applications. In *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference* (2004), pp. 21–30.